*remote sensing*

MDPI

*Article*

# Selecting Post-Processing Schemes for Accurate Detection of Small Objects in Low-Resolution Wide-Area Aerial Imagery

**Xin Gao [1,*], Sundaresh Ram [2], Rohit C. Philip [3], Jeffrey J. Rodríguez [1], Jeno Szep [4], Sicong Shao [1], Pratik Satam [1], Jesús Pacheco [5] and Salim Hariri [1]**

[1] Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721, USA; jrod@ece.arizona.edu (J.J.R.); sicongshao@email.arizona.edu (S.S.); pratiksatam@email.arizona.edu (P.S.); hariri@ece.arizona.edu (S.H.)
[2] Department of Radiology, Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI 48109, USA; sundarer@med.umich.edu
[3] Genetesis, Mason, OH 45040, USA; rphilip@email.arizona.edu
[4] Fraunhofer USA Center Mid-Atlantic (CMA), Riverdale, MD 20737, USA; jeno.szep@gmail.com
[5] Department of Industrial Engineering, Universidad de Sonora, Hermosillo 83000, Mexico; jesus.pacheco@unison.mx
* Correspondence: tzgaoxin2016@hotmail.com or xgao1985@email.arizona.edu; Tel.: +1-(520)-304-6028

**Abstract:** In low-resolution wide-area aerial imagery, object detection algorithms are categorized as feature extraction and machine learning approaches, where the former often requires a post-processing scheme to reduce false detections and the latter demands multi-stage learning followed by post-processing. In this paper, we present an approach on how to select post-processing schemes for aerial object detection. We evaluated combinations of each of ten vehicle detection algorithms with any of seven post-processing schemes, where the best three schemes for each algorithm were determined using average F-score metric. The performance improvement is quantified using basic information retrieval metrics as well as the classification of events, activities and relationships (CLEAR) metrics. We also implemented a two-stage learning algorithm using a hundred-layer densely connected convolutional neural network for small object detection and evaluated its degree of improvement when combined with the various post-processing schemes. The highest average F-scores after post-processing are 0.902, 0.704 and 0.891 for the Tucson, Phoenix and online VEDAI datasets, respectively. The combined results prove that our enhanced three-stage post-processing scheme achieves a mean average precision (mAP) of 63.9% for feature extraction methods and 82.8% for the machine learning approach.

**Keywords:** post-processing; vehicle detection; wide-area aerial imagery; segmentation; machine learning



**Citation:** Gao, X.; Ram, S.; Philip, R.C.; Rodríguez, J.J.; Szep, J.; Shao, S.; Satam, P.; Pacheco, J.; Hariri, S. Selecting Post-Processing Schemes for Accurate Detection of Small Objects in Low-Resolution Wide-Area Aerial Imagery. *Remote Sens.* **2022**, *14*, 255. https://doi.org/10.3390/rs14020255

Academic Editors: Jan Platoš and Dusan Gleich

Received: 20 October 2021
Accepted: 29 December 2021
Published: 6 January 2022



## 1. Introduction

In object detection, small objects are regarded as those taking up less than 1% of an image when measuring the area of objects with a bounding box [1]. Detecting small objects in low-resolution wide-area aerial imagery is a challenging task, especially in remote sensing applications [2–15]. For example, in widely recognized datasets such as VEDAI, DOTA, MS COCO, Potsdam, and Munich [1,16–18] there is an interest in locating small objects of interest that have a size of less than 32 × 32 pixels. In low-resolution aerial datasets, small objects such as cars or trucks usually cover 20 to several hundred pixels in area, which renders the detection task in a sample aerial video rather difficult. Detecting small objects in an aerial video is of importance in many practical applications including urban traffic management, visual surveillance, local parking monitoring, military target strike, and emergency rescue [3,4,12,14,16,18]. Widely recognized as a special case of object detection, detecting small objects in aerial imagery is challenging due to issues such as high density of the objects within a small area, shadow from clouds, partial occlusion due to

other objects, and varying orientations [19]. In addition, factors such as model diversity, illumination change, low contrast, and low resolution, have made it rather difficult for recent object detection algorithms to achieve accurate detection results [20,21]. Given the relatively high number of false detections in many object detection algorithms in low-resolution aerial video, we have studied a few popular post-processing schemes [4,13,14,20] to help improve the object detection results, while it is also vital on selecting a decent post-processing method for feature extraction and machine learning approaches.

We have presented the performance analysis of six object detection algorithms in our prior work [2] and studied some existing post-processing schemes [4,22–25], to improve the performance of each algorithm. Salem et al., presented the filtered dilation scheme [22] with a median filter for smoothing, followed by a dilation operator to expand the shapes contained in binary images. Samarabandu and Liu used a heuristic filtering scheme [23] (with two adjustable thresholds for area and aspect ratio) to discard false detections after extracting 8-connected regions. Sharma et al., applied the metric of shape index [24] to filter out false detections less than the smallest shape index. Zheng et al., developed the sieving and opening scheme [25] for removing detected objects which are too large or too small to be a vehicle. Motivated by the four existing post-processing schemes [22–25], we have derived a two-stage scheme (sieving and closing [3,4]), which only retains vehicle size within an area range ($[t_{\text{low}}, t_{\text{high}}]$) followed by a morphological closing operation. We also proposed a three-stage scheme [13], which applies area thresholding followed by morphological closing and conditional sieving based on a compactness measure [26].

We evaluated seven post-processing schemes applied to ten feature extraction-based algorithms, using average F-score as the primary evaluation metric. In the first set of experiments, we measured the performance of filtered dilation (FiltDil) [22], heuristic filtering (HeurFilt) [23], sieving and opening (S&O) [25], sieving and closing (S&C) [3,4], the three-stage (3Stage) scheme [13], the enhanced three-stage (Enh3Stage) scheme [20] and spatial processing scheme (SpatialProc) [14] applied to the ten algorithms [25,27–35] adapted for aerial vehicle detection. Randomly grouped frames were used for preliminary tests using two aerial image datasets (Tucson and Phoenix). We then determined the best three post-processing schemes according to overall average F-score for each algorithm. We classified two types of detection errors for the two datasets: the first type includes false negative (FN or Miss) and false positive (FP), the second type includes Splits (S) and Merges (M). The second set of experiments determined the best post-processing scheme for traditional approaches for small object detection in aerial image datasets. In the third set of experiments, we investigated the selection of post-processing schemes to combine with an 11th algorithm, which was a machine learning approach for vehicle detection. Inspired by some similar approaches such as spatial analysis [36] and smoke vehicle detection [37,38], besides the ten detection algorithms [25,27–35], we adapted a two-stage machine learning approach from the tiramisu code [39] for semantic segmentation to perform the classification task via a hundred-layer densely connected convolutional network (DenseNets). The three post-processing schemes [4,14,20] that performed best in the first set of experiments were each applied to the machine learning approach. To assess the performance, we used a third aerial dataset (online VEDAI) to determine the optimal selection of post-processing. Posterior tests were conducted using Google Colab Pro, where the speed-up ratio for each algorithm was also evaluated.

Our quantitative results were obtained via two sets of evaluation metrics: basic information retrieval (IR) metrics [40] and the CLEAR metrics [41,42], where the former includes precision, recall, F-score, accuracy and percentage of wrong classification (PWC), and the latter comprises multiple object detection accuracy (MODA) and multiple object counts (MOC). For the fairness of comparing each detection to the ground truth (GT), we did not adopt the metric of multiple object detection precision [19] since the selected post-processing methods did not implement temporal analysis, and the eleven algorithms we adapted on aerial vehicle detection [25,27–35,39] lack spatial overlap information.

The major contributions of our research study are summarized below:

(i) We designed three sets of experiments to measure the degree of performance improvement on each of the seven post-processing schemes for any of the eleven algorithms on vehicle detection. The first set of experiments determined the best three post-processing schemes using two aerial image datasets (Tucson and Phoenix). The second set of experiments decided the best post-processing scheme for each of the ten algorithms. The third set of experiments conducted verification using two groups of selected post-processing methods for the 11th algorithm on a third public aerial dataset (online VEDAI).

(ii) We adapted a two-stage learning scheme by applying DenseNet to perform the detection task and combined our post-processing scheme to improve object detection accuracy. Quantitative results proved the viability and efficiency for small object detection using a low-resolution wide-area aerial dataset, i.e., the online VEDAI dataset, which is not limited to small cars and trucks.

(iii) We measured the degree of improvement on object detection performance in addition to a time-savings evaluation. We adjusted the parameters for learning and validation to achieve the highest accuracy without any post-processing, then applied the selected post-processing and measured the speed-up factor using Google Colab Pro.

The remainder of this paper is organized as follows. In Section 2, we present some related work on traditional and current state-of-the art schemes for small object detection in the remote sensing domain. In Section 3, we present a concise summary of the existing schemes as well as our proposed schemes for post-processing, and some key details on DenseNet, in addition to its architecture adapted for two-stage learning. Section 4 includes a discussion of our experiments along with quantitative and qualitative results. Section 5 discusses the results of our research study. Finally, Section 6 presents concluding remarks and future research directions.

## 2. Algorithms Adapted for Vehicle Detection

Motivated by the previously published research articles on object detection and image segmentation [25,27–35], we adapted ten feature extraction-based algorithms for the purpose of detecting small cars or trucks on roadways. The ten algorithms we studied include the localization contrast model (LC) [27], the variational minimax optimization (VMO) based algorithm [28], frequency toned (FT) approach [29], maximum symmetric surround saliency (MSSS) [30], a hybrid of kernel-based fuzzy c-means (KFCM) and Chan-Vese (CV) model [31], the partial area effects (PAE) scheme [34], feature density estimation (FDE) [35], morphological filtering (MF) [25], text extraction (TE) [32] and the fast independent component analysis (FICA) method [33]. We summarized the crucial details of each algorithm in [4] upon adapting each algorithm for parameter tuning to achieve the highest average F-score on each dataset. The most suitable algorithm parameters were determined via the best overall average F-score for these ten algorithms [25,27–35] each combined with the four existing post-processing schemes [4,22,23,25], as well as the 3Stage [13], Enh3Stage [20], and SpatialProc [14].

To compare the degree of performance improvement of post-processing schemes for the VMO-based algorithm [28], we applied the three-stage scheme [13] in contrast to each of five post-processing schemes [4,22–25], and analyzed their performance. In our first set of experiments, each of the ten algorithms [25,27–35] was combined with each of the post-processing schemes [4,13,14,20,22,23,25] to determine the best three post-processing schemes. The second set of experiments determined the best post-processing scheme for traditional approaches for small object detection in aerial image datasets. A third set of experiments assessed the hundred-layer tiramisu code using DenseNet [39] with learning parameters optimized for the online VEDAI dataset, in combination with various post-processing schemes.

Given the fact that the established fully convolutional (FC) DenseNet trained their models with neither extra data nor post-processing, we adapted this scheme with initial learning rates, performed similar regularization [39], and enhanced the global accuracy using our selected post-processing. We also studied several other recent schemes for small

object detection. Li et al. proposed an improved two-stage detector CANet based on faster region-based convolutional neural network (R-CNN), which effectively enriched context information and nonlocal information when detecting small objects [43], but they did not report the time efficiency and the outcome of mean average precision (mAP). Another VGG-16-based method for multi-object detection, showed improved accuracy in two manually labeled datasets, but the work admits its limitations on more complex traffic scenes and suggests generative adversarial networks as an alternative plan to solve this task [44]. The spatio-temporal ConvNet aims for video-based small object detection, however, it may not be suitable for a limited number of small objects or very large-scale datasets when each frame may require independent manual annotations [45]. Wang and Gu's framework [46] for small target detection combined the feature-balanced pyramid network (FBPN) with faster R-CNN. Gao and Tian [47] presented a generative adversarial network specifically designed for weak vehicle detection in satellite images. While these approaches [43–47] and other deep CNN strategies [48,49] have been proposed to find a better solution for the task of small-object detection, they lack a post-processing module. This module neither appeared in the study of relationship between accuracy and the size of structuring element [50], nor the sliding-window based approach on vehicle detection [51] according to our previous study.

While our prior manual segmentation was performed on the two low-resolution aerial video datasets (Tucson and Phoenix), we did not use the dual-threshold technique for temporal analysis [14] for any of the three aerial image datasets. Therefore, all the algorithms were performed on each frame independently.

## 3. Post-Processing Schemes

We previously derived a two-stage post-processing method sieving and closing (S&C) [4]) to reduce false detections. In the first stage, we perform area thresholding which sieves out any detected objects whose area fall outside a designated range. We used a pixel-area range of (5, 160) for the Tucson dataset and (5, 180) for the Phoenix dataset (because the spatial resolutions are slightly different). In the second stage, due to the persistence of some detection errors even after applying area thresholds, a morphological closing operation was performed to connect adjacent tiny objects which tend to be false detections. This step also offers boundary smoothing and fills small holes inside each detection. The size of a morphological closing filter [50] is flexible with respect to the criteria of achieving highest overall average F-score. All the binary objects within an area range encompassing most small cars and trucks are preserved, but fixed thresholding does not generalize to datasets with different resolutions [13]. Inspired by sieving and closing [4], we derived the three-stage scheme for post-processing [13], where Stage 1 operates area thresholding to drop some incorrect detections, Stage 2 performs the morphological closing operation and Stage 3 applies conditional object sieving with respect to a compactness measure for vehicle shape. This scheme overcame the shortcomings of using expected vehicle size and fit datasets with various spatial resolutions [13]. In Stage 3, the compactness $C$ of a region is defined as [26]

$$C = \frac{L^2}{4\pi A} \tag{1}$$

where $L$ represents the perimeter of the region, and $A$ is the area of the region. For our data, we define the lower compactness threshold as half of the smallest compactness, and the upper threshold as twice the largest compactness. Due to the concern of distortions on some small vehicles at high probability to be truly detected objects, we retained detections with compactness in the range $[C_{\text{small}}/2, 2 \times C_{\text{large}}]$ [4,13].

The enhanced three-stage post-processing scheme [20] was derived to achieve more independence with respect to object size [33]. In Stage 1, a $3 \times 3$ median filter is applied. In Stage 2, an opening operation is applied to sieve out trivial false detections, followed by a closing operation. Finally, Stage 3 applies linear Gaussian filtering followed by non-maximum suppression (NMS) to discard multiple false detections nearby a single object.

We also developed a two-stage spatial processing scheme [14], where the first stage used multi-neighborhood hysteresis thresholding. The second stage of spatial processing is identical to Stage 2 of the Enh3Stage post-processing scheme [20], while the filter size of the opening and closing operation was carefully adjusted in accordance with the spatial resolution of the datasets.

## 4. Experiments and Results

We evaluated quantitative detection performance in four scenarios: (i) six detection algorithms combined with the 3Stage scheme and each of these six with filtering by shape index (SI); (ii) seven post-processing schemes each associated with any of the ten object detection algorithms in the Tucson and Phoenix datasets to determine the best three post-processing schemes for each algorithm; (iii) ten algorithms each combined with three post-processing schemes out of seven as determined from (ii), along with a visual comparison of the vehicle detection results; (iv) a two-stage machine learning approach with adjusted parameters to achieve the highest overall accuracy on small object detection in the online VEDAI dataset, using its updated flowchart with an optimally tuned proportion of folders on training to testing and updating its structure of FC-DenseNet, combined with each of the three post-processing schemes determined from (iii). Figure 1 depicts a pipeline of our research study on selecting post-processing schemes for the detection of small objects.

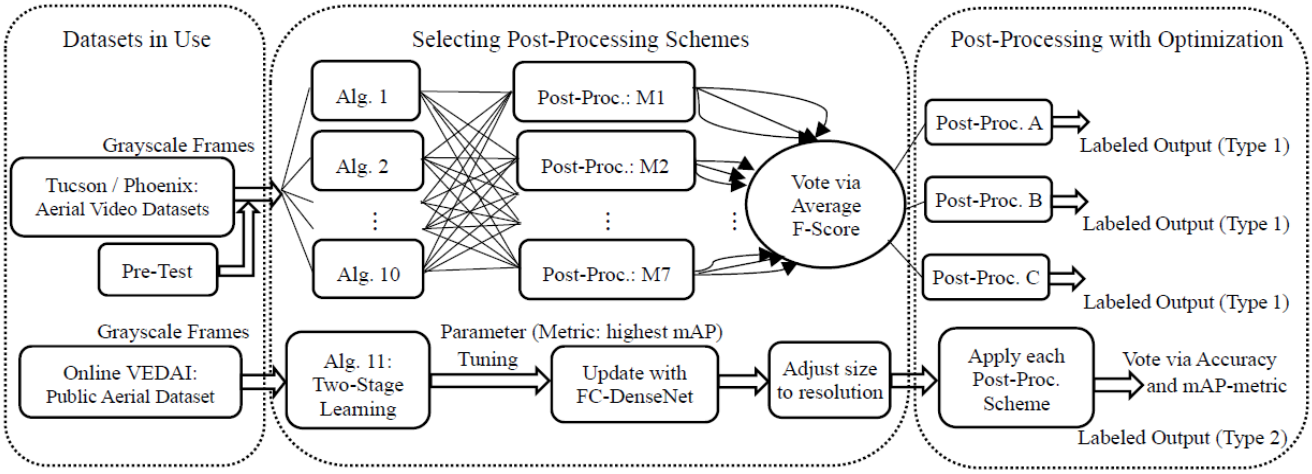


**Figure 1.** Pipeline of the proposed research study of small object detection. Labeled outputs of Type 1 are detection results of using ten feature extraction algorithms followed by each of the seven post-processing schemes. Labeled outputs of Type 2 correspond to an updated multi-stage learning approach (FC-DenseNet) with selected post-processing schemes.

### *4.1. Experimental Setup*

We conducted our experiments using MATLAB R2019b on a Windows PC (Intel Core i7-8500U, 1.80 GHz CPU, 16 GB RAM). The average computation time per frame for each detection algorithm was found by calculating the total processing time of each algorithm before and after combining the selected post-processing scheme divided by the total number of frames in the experiment. Meanwhile, we applied the vehicle tiramisu code (Python version) [39] for image segmentation in the Google Colab Pro environment; after saving the outputs in a .mat file, we implemented our post-processing code in MATLAB for each of the three datasets.

Two versions were implemented for each of the ten detection algorithms [14]: (1) using our prior binarization methods [4,20] for simple thresholding, and (2) replacing the binarization step with the proposed spatial processing. We refer to these two approaches as "before" and "after" using a post-processing scheme in our experimental analysis, respectively. We compare the labeled outputs of the "before" and "after" to

quantify the degree of improvement of object detection performance resulting from the proposed post-processing.

### *4.2. Datasets*

Two aerial videos (spatial resolution: 720 × 480 pixels per frame) obtained from a low-resolution camcorder, served as our datasets for performance analysis of the ten vehicle detection algorithms [25,27–35] each associated with any of the five post-processing schemes [4,22–25], the three-stage scheme by the measure of compactness [13], and the enhanced three-stage scheme [20]. In our manual segmentation, the total number of ground truth (GT) vehicles is 8072 (4012 in the Tucson dataset and 4060 in the Phoenix dataset). All these vehicles have approximately rectangular shape, and the vehicle area is distributed from 40 to 150 pixels in the Tucson dataset and ranged from 20 to 175 pixels in the Phoenix dataset. Automatic detections from any combination of algorithms without or with a post-processing scheme are compared with their GT vehicles in each frame.

We used online VEDAI dataset [52] as a third dataset (spatial resolution: 512 × 512 pixels per frame), which comprises 9 classes of objects; except for buses, boats, and planes, this study concerns small size vehicles such as cars, pickups, tractors, camping cars, trucks and vans [46]. The online VEDAI dataset contains 1246 images with a total of 3600 instances across these 6 classes. The portion of this dataset used for training was varied from 80% to 95% in increments of 5%, then 98% as highest; the rest were used for testing.

### *4.3. Classifications and Evaluation Metrics*

We evaluated each algorithm [25,27–35] for small object detection by automatically classifying the correct detection and each type of detection errors. The binary detection outputs were processed with 8-connected component labeling, and the overlaps between detections and the ground truth, were evaluated similar to the region matching proposed by Nascimento and Marques [53], in which the detections are characterized as follows:

- True positive (TP): correct detection. Only one TP (which has the largest overlap with ground truth if exists) is counted if multiple detections intersect the same ground truth object, or a single detection intersects multiple ground truth objects.
- Splits (S): if multiple detections touch a single object in the ground truth, then only one TP (the one having largest overlap) is counted, all other touches are regarded as Splits.
- Merges (M): if a single detection is associated with multiple objects in the ground truth, then all other objects except the TP in a row are counted as Merges.
- False negative (FN) or Miss: detection failure, indicated by a ground truth object that fails to intersect any detection.
- False positive (FP): incorrect detection, indicated by a detection which fails to intersect any ground truth object.

To quantify the performance of each algorithm with and without post-processing, we use basic information retrieval (IR) metrics [3,4,40]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$F_\beta = \frac{(1+\beta^2)\text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}} \tag{4}$$

When $\beta = 1$, the $F_\beta$ measure equals $F_1$-score, a harmonic mean of precision and recall. Substituting precision and recall in the expression of TP, FP and FN into Equation (4), the simplified expression of $F_1$-score can be written as

$$F_1\text{-score} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{5}$$

Percentage of wrong classification (PWC) measures the ratio of FP and FN in total, to the sum of TP, FP, FN and TN [17]

$$\text{PWC} = \frac{\text{FN} + \text{FP}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{6}$$

An algorithm with lower PWC score indicates better detection results.

To assess comprehensive performance of each algorithm before and after combining with a post-processing scheme, we adopted the CLEAR metrics [41,42] for quantitative evaluations. If we denote the number of Misses (FNs) as $m_i$ and the number of FPs as $fp_i$, then the multiple object detection accuracy (MODA) in the $i$-th frame ($i = 1, \ldots, 100$ in each dataset) is computed as [41]

$$\text{MODA}(i) = 1 - \frac{c_m(m_i) + c_f(fp_i)}{N_G^{(i)}} \tag{7}$$

where $c_m$ and $c_f$ represent the weights applied to the FNs and FPs, respectively; $N_G^{(i)}$ is the number of ground-truth objects in the $i$-th frame.

We equally weight $c_m = c_f = 1$ [41] and sum over all frames in each dataset [42], yielding the multiple object count (MOC)

$$\text{MOC} = 1 - \frac{\sum_{i=1}^{N_{frames}} [(m_i) + (fp_i)]}{\sum_{i=1}^{N_{frames}} N_G^{(i)}} \tag{8}$$

Since no negative sample exists in the ground truth of the Tucson and Phoenix datasets, the detection outputs for these datasets do not contain any true negatives (TN). Meanwhile, Splits (S) or Merges (M) were a second type of detection error; they were not counted in either MODA or MOC. There has not been any final agreement on how to weight Splits (S) or Merges (M) in the previous publications [40,42,53,54].

In online VEDAI dataset, we used the accuracy metric to evaluate the object detection performance, which is the ratio of the number of correct predictions to the total number of predictions [55,56]. For binary classification, the accuracy metric equals the percentage of correctly classified instances, which is expressed as [40]

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{9}$$

where the accuracy metric numerically matches 1−PWC.

Average precision is defined as [57–59]

$$\text{AP} = \int_0^1 P(R)\, dR \tag{10}$$

where P = Precision and R = Recall. The mean average precision (mAP) is a global performance measure defined as the sample mean of AP [59]

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{11}$$

### *4.4. Results and Analysis*

#### 4.4.1. Six Vehicle Detection Algorithms Combined with our Three-Stage Post-Processing Scheme or Filtering by SI

We started our experimental study by evaluating six algorithms [27–32] combined with the 3Stage scheme [13] or filtering by SI [24]. We carefully adjusted the parameters of each algorithm to achieve the highest average F-score. Table 1 displays the average

F-score of each method for two datasets. Table 1 shows that our 3Stage scheme has better average F-score than filtering by SI for all six algorithms; hence, we decided to replace filtering by SI with our 3Stage scheme in subsequent experiments involving combinations of post-processing schemes with each of the ten detection algorithms.

**Table 1.** Average F-score for three-stage post-processing scheme vs. filtering by SI.

| Alg. | Tucson Dataset | | Phoenix Dataset | |
|---|---|---|---|---|
| | **3Stage Scheme [13]** | **Filtering by SI [24]** | **3Stage Scheme [13]** | **Filtering by SI [24]** |
| **LC** [27] | 0.860 | 0.736 | 0.625 | 0.482 |
| **FT** [29] | 0.824 | 0.716 | 0.478 | 0.409 |
| **MSSS** [30] | 0.724 | 0.620 | 0.450 | 0.385 |
| **VMO** [28] | 0.775 | 0.551 | 0.704 | 0.501 |
| **TE** [32] | 0.643 | 0.489 | 0.548 | 0.436 |
| **CV** [31] | 0.584 | 0.523 | 0.401 | 0.257 |

#### 4.4.2. Seven Post-Processing Schemes Combined with Ten Algorithms

We randomly divided the 100 frames in each of the Tucson and Phoenix datasets into ten groups where each group has ten frames. Post-processing (3Stage scheme [13], sieving and closing (S&C) [4]) combined with LC [27], VMO [28] and FDE [34] were selected to calculate average F-scores via 100 frames in each of the two datasets. These groups of average F-scores were obtained where each group contains ten frames. Table 2 shows the mean and standard deviation for each group, along with the 95% confidence intervals (CI); each group of data has tight CI along with considerably small standard deviation.

**Table 2.** Statistical test on average F-score (10 × 10 frames): two post-processing schemes associated with three automatic detection algorithms.

| Alg. | Tucson Dataset | | Phoenix Dataset | |
|---|---|---|---|---|
| | **3Stage Scheme [13]** | **S&C [4]** | **3Stage Scheme [13]** | **S&C [4]** |
| **LC** [27] | 0.839 ± 0.006 (Std. = 0.008) | 0.871 ± 0.005 (Std. = 0.007) | 0.589 ± 0.010 (Std. = 0.014) | 0.620 ± 0.009 (Std. = 0.012) |
| **VMO** [28] | 0.745 ± 0.014 (Std. = 0.019) | 0.802 ± 0.009 (Std. = 0.012) | 0.696 ± 0.007 (Std. = 0.010) | 0.683 ± 0.014 (Std. = 0.020) |
| **FDE** [34] | 0.871 ± 0.004 (Std. = 0.006) | 0.876 ± 0.005 (Std. = 0.007) | 0.658 ± 0.013 (Std. = 0.018) | 0.696 ± 0.009 (Std. = 0.013) |

We evaluated the ten feature extraction-based vehicle detection algorithms combined with each of the seven post-processing methods [4,13,14,20,22,23,25] (marked as M1 to M7) for all the 200 frames in the Tucson and Phoenix datasets. The average F-scores are tabulated in Table 3, where the best three overall F-scores (boldface numbers in each column) were selected for further comparison.

From Table 3, we count the number of boldface entries in each row to determine that the Enh3Stage scheme (M6) obtained the highest nine votes, S&C (M5) and SpatialProc (M7) both obtained six votes, 3Stage scheme (M3) had four votes, and each of the other three (M1, M2, M4) was voted no more than twice.

**Table 3.** Average F-score for seven post-processing methods combined with each of the ten algorithms for Tucson (T) and Phoenix (P) datasets.

| Post-Proc. \ Alg. | | LC [27] | VMO [28] | FT [29] | MSSS [30] | KFCM-CV [31] | FDE [34] | PAE [35] | MF [25] | TE [32] | FICA [33] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Method 1 (M1): FiltDil [22] | T | 0.892 | 0.760 | **0.848** | **0.782** | 0.725 | 0.879 | 0.625 | 0.681 | 0.640 | 0.743 |
| | P | 0.549 | 0.714 | **0.608** | **0.631** | 0.306 | 0.643 | 0.313 | 0.454 | 0.621 | 0.392 |
| Method 2 (M2): HeurFilt [23] | T | 0.840 | 0.717 | 0.766 | 0.724 | 0.569 | 0.867 | 0.801 | 0.724 | **0.812** | **0.804** |
| | P | 0.538 | 0.606 | 0.453 | 0.354 | 0.296 | 0.591 | 0.480 | 0.380 | **0.525** | **0.377** |
| Method 3 (M3): 3Stage [13] | T | **0.860** | **0.775** | 0.824 | 0.724 | 0.584 | **0.866** | **0.793** | 0.618 | 0.643 | 0.720 |
| | P | **0.625** | **0.704** | 0.478 | 0.450 | 0.401 | **0.673** | **0.646** | 0.341 | 0.548 | 0.337 |
| Method 4 (M4): S&O [25] | T | 0.822 | 0.515 | 0.733 | 0.641 | 0.467 | 0.877 | 0.798 | **0.776** | 0.815 | 0.655 |
| | P | 0.583 | 0.640 | 0.439 | 0.470 | 0.205 | 0.651 | 0.535 | **0.588** | 0.498 | 0.355 |
| Method 5 (M5): S&C [4] | T | **0.881** | **0.801** | 0.861 | 0.776 | **0.587** | **0.885** | **0.814** | **0.722** | 0.680 | 0.723 |
| | P | **0.628** | **0.702** | 0.539 | 0.568 | **0.407** | **0.683** | **0.579** | **0.500** | 0.565 | 0.433 |
| Method 6 (M6): Enh3Stage [20] | T | 0.823 | **0.820** | **0.832** | **0.817** | **0.610** | **0.902** | **0.762** | **0.683** | **0.754** | **0.712** |
| | P | 0.611 | **0.688** | **0.598** | **0.592** | **0.479** | **0.614** | **0.627** | **0.518** | **0.642** | **0.461** |
| Method 7 (M7): SpatialProc [14] | T | **0.842** | 0.691 | **0.809** | **0.808** | **0.694** | 0.857 | 0.723 | 0.764 | **0.784** | **0.738** |
| | P | **0.646** | 0.544 | **0.547** | **0.572** | **0.346** | 0.632 | 0.534 | 0.482 | **0.557** | **0.426** |

#### 4.4.3. Ten Algorithms Combined with the Best Three Post-Processing Schemes

We step further to present the performance analysis before and after post-processing, where the types of detections were analyzed for all the 8072 vehicles in the Tucson and Phoenix datasets. Table 4 shows the quantitative results on each of the ten algorithms before and after combining with each of their best three post-processing schemes (as highlighted with bold numbers in Table 3). Note that M0 refers to no post-processing applied to an algorithm in Table 4 and the subsequent tables.

**Table 4.** Types of detections for each algorithm combined with each of the best three schemes from Table 3.

| Alg. | Scheme | Counts of Detection Types | | | | | Alg. | Scheme | Counts of Detection Types | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | TP | S | M | FN | FP | | | TP | S | M | FN | FP |
| LC [27] | **M0** | **7051** | 1631 | 688 | 333 | 7691 | FDE [34] | M0 | 6477 | 1677 | 362 | 1233 | 6703 |
| | M3 | 5862 | 241 | 410 | 1800 | 3110 | | M3 | 5644 | 267 | 519 | 1909 | 1648 |
| | M5 | 6187 | 368 | 345 | 1540 | 3792 | | M5 | 5447 | 150 | 974 | 1651 | 1347 |
| | M7 | 5988 | 123 | 415 | 1669 | 2327 | | M6 | 5823 | 146 | 512 | 1689 | 2055 |
| VMO [28] | M0 | 7104 | 10,527 | 188 | 584 | 16,633 | PAE [35] | M0 | 6977 | 6581 | 81 | 1014 | 14,416 |
| | M3 | 5585 | 653 | 853 | 1634 | 3065 | | M3 | 5566 | 134 | 627 | 1879 | 2552 |
| | M5 | 5920 | 746 | 973 | 1179 | 3619 | | M5 | 5807 | 373 | 497 | 1768 | 3198 |
| | M6 | 6247 | 813 | 671 | 1154 | 2890 | | M6 | 5703 | 231 | 467 | 1902 | 3127 |
| FT [29] | M0 | 7290 | 5000 | 174 | 608 | 14,888 | MF [25] | M0 | 6753 | 10,100 | 73 | 1246 | 37,742 |
| | M1 | 6357 | 1879 | 669 | 1046 | 5667 | | M4 | 5056 | 1562 | 102 | 2914 | 2568 |
| | M6 | 6076 | 625 | 285 | 1711 | 3070 | | M5 | 5125 | 322 | 1529 | 1418 | 6140 |
| | M7 | 5541 | 892 | 286 | 2245 | 2769 | | M6 | 4994 | 240 | 277 | 3001 | 3280 |
| MSSS [30] | M0 | 7457 | 5907 | 353 | 262 | 17,286 | TE [32] | M0 | 6755 | 6264 | 191 | 1126 | 25,879 |
| | M1 | 6394 | 2005 | 1049 | 629 | 4772 | | M2 | 5631 | 1966 | 183 | 2258 | 3357 |
| | M6 | 6133 | 543 | 264 | 1675 | 3413 | | M6 | 5668 | 1607 | 104 | 2300 | 2575 |
| | M7 | 5906 | 555 | 323 | 1843 | 3338 | | M7 | 5597 | 1560 | 93 | 2382 | 2401 |
| KFCM-CV [31] | M0 | 7111 | 12,053 | 428 | 533 | 41,539 | FICA [33] | M0 | 6766 | 4263 | 98 | 1208 | 32,360 |
| | M5 | 5212 | 1577 | 443 | 2417 | 9697 | | M2 | 5426 | 880 | 18 | 2628 | 5262 |
| | M6 | 5294 | 1287 | 553 | 2225 | 6950 | | M6 | 5227 | 419 | 230 | 2615 | 3991 |
| | M7 | 5145 | 1356 | 425 | 2502 | 8738 | | M7 | 5545 | 189 | 604 | 1923 | 6660 |

As observed from Table 4, among nine feature extraction-based algorithms combined with Enh3Stage scheme (M6), VMO and MSSS display the highest and second highest TP

counts; among six algorithms with SpatialProc (M7), LC and MSSS show the highest and second highest TP counts; among five algorithms with S&C (M5), LC and VMO represent the best two in TP counts. Among the tabulated ten detection algorithms combining any type of post-processing, MSSS with FiltDil (M1) presents the largest TP counts, while MF with S&O (M4) shows the lowest TP counts. Regarding FNs (Misses), MSSS with FiltDil (M1) and VMO with Enh3Stage scheme (M6) show the smallest and second smallest FN counts. Regarding FP, the lowest FP count was found in FDE with S&C (M5), the best reduction of FP was shown by MF with S&O (M4), displaying 93.2% decrease of FP, the least reduction was found on LC with S&C (M5), displaying 50.7% decrease on FP. Meanwhile, all the post-processing schemes in Table 4 reduced Splits (S), where the best reduction was achieved by PAE with 3Stage scheme (M3), and the least reduction was obtained by FT with FiltDil (M1). All seven post-processing schemes for any of the ten algorithms trade improving average F-score for the mild cost of converting a small part of TPs to Merges. For reducing FPs and Splits, each post-processing scheme has quite similar performance when detecting these small vehicles using our aerial datasets.

We used basic IR metrics [40] to evaluate the performance of ten detection algorithms before and after applying post-processing. The average precision, recall and F-score of each algorithm before using any post-processing are depicted in Figure 2. Regarding precision and F-score, the highest values were achieved by LC in Tucson dataset, while the lowest corresponded to CV (KFCM-CV) in the Tucson dataset and FICA in the Phoenix dataset. Regarding recall, all the rates of eight algorithms (except FDE and PAE) are higher than 0.9 in the Tucson dataset, while only two algorithms (LC and MSSS) achieved recall rates higher than 0.9 in the Phoenix dataset.

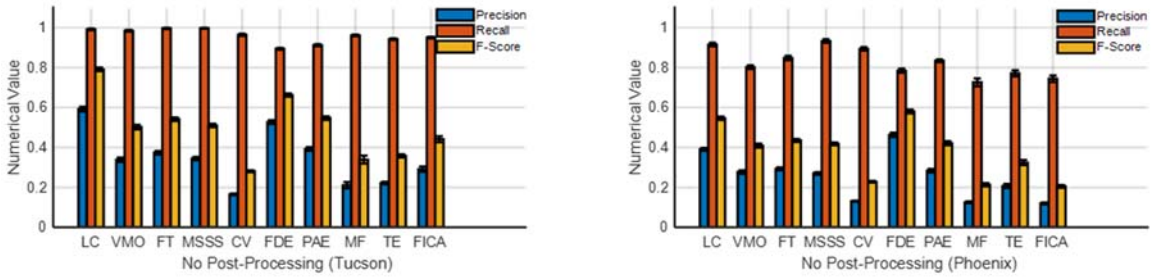


**Figure 2.** Precision, Recall and F-score of ten detection algorithms for (**a**) Tucson dataset and (**b**) Phoenix dataset.

The detection results of ten algorithms associated with the voted best three post-processing schemes were arranged into four groups: those algorithms with Enh3Stage scheme (M6), those with S&C (M5) and SpatialProc (M7), and those with all other four schemes (M1 through M4). The precision, recall and F-score on each of the ten algorithms after combining with the best three post-processing schemes are displayed in Figure 3, which consists of four sub-diagrams. The results for the two datasets were merged to simplify the visual comparison in Figure 3. Regarding precision and F-score, when post-processed by the Enh3Stage scheme (M6), FDE and CV ranked the highest and lowest among the eight algorithms, respectively; the same conclusion was drawn when post-processed by S&C (M5), while the highest and lowest results were displayed with LC and CV when post-processed by SpatialProc (M7). For other post-processing schemes, the highest scores were achieved by FDE with 3Stage scheme (M3), the lowest scores were obtained by FICA with HeurFilt (M2). Regarding recall, seven algorithms (except MF and FICA) preserved their recall rates above 0.7 when post-processed by the Enh3Stage scheme (M6) or any of the other four schemes (M1 through M4); five algorithms (except CV) retained their recall rates above 0.7 when post-processed by either S&C (M5) or SpatialProc (M7). Three of the best precision and average F-scores were achieved by FDE with S&C

(M5), FDE with 3Stage scheme (M3), and FDE with Enh3Stage scheme (M6), while the best three recalls were achieved by MSSS with FiltDil (M1), VMO with S&C (M5), and VMO with the Enh3Stage scheme (M6).

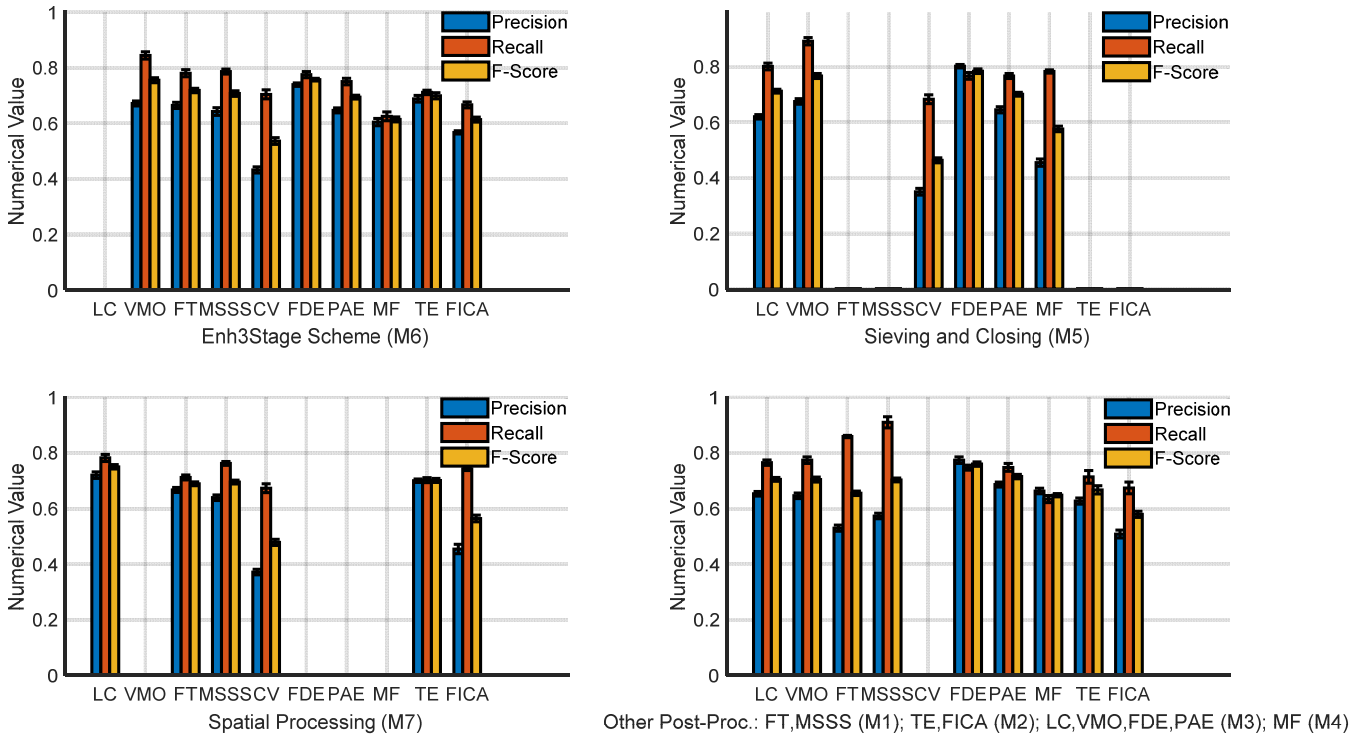


**Figure 3.** Precision, Recall and F-score for ten feature extraction-based algorithms each combined with their best three post-processing schemes.

As seen in Figure 3, the improvements on precision and average F-scores of all ten vehicle detection algorithms supported the validity of our post-processing selection, while the mild decrease of recall for each algorithm resulted from the trade-off of losing a small portion of TPs. Due to the extremely low-resolution of wide-area aerial frames, none of the algorithms combined with post-processing achieved an average F-score higher than 0.9.

Table 5 displays the percentage of wrong classification (PWC) for our quantitative results on ten feature extraction-based algorithms with their three best post-processing schemes. The best outcome was achieved by FDE with the Enh3Stage scheme (M6), which resulted in a PWC of 18.2% in Tucson dataset. The smallest improvement was obtained by LC with S&C (M5), where PWC reduced from 62.4% to 58.9% in the Phoenix dataset. In the Tucson dataset, the PWC scores of nine algorithms (excluding KFCM-CV) were reduced to below 50%; in the Phoenix dataset, only VMO with any of its three best post-processing schemes (M3, M5 and M6) and FDE with S&C (M5), decreased to below 50%. Regarding the degree of improvement, the best three post-processing schemes for each of nine algorithms (except VMO) performed better in the Tucson dataset than in the Phoenix dataset. In contrast, only VMO when post-processed by the 3Stage scheme (M3) or S&C (M5), exhibited better scores in the Phoenix dataset than those in the Tucson dataset.

**Table 5.** PWC for ten algorithms before and after combining with their best three post-processing schemes.

| Alg. \ Scheme | PWC %/Tucson Dataset | | | | | | | | PWC %/Phoenix Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **M0** | **M1 [22]** | **M2 [23]** | **M3 [13]** | **M4 [25]** | **M5 [4]** | **M6 [20]** | **M7 [14]** | **M0** | **M1 [22]** | **M2 [23]** | **M3 [13]** | **M4 [25]** | **M5 [4]** | **M6 [20]** | **M7 [14]** |
| **LC [27]** | 41.5 ± 1.1 | | | 27.6 ± 1.2 | | 27.9 ± 1.2 | | 27.3 ± 1.0 | 62.4 ± 0.7 | | | 58.4 ± 1.0 | | 58.9 ± 1.0 | | 52.3 ± 1.1 |
| **VMO [28]** | 66.3 ± 1.1 | | | 40.8 ± 1.1 | | 46.9 ± 1.0 | 35.0 ± 1.2 | | 74.2 ± 0.8 | | | 46.9 ± 1.0 | | 49.8 ± 1.2 | 47.6 ± 1.1 | |
| **FT [29]** | 62.7 ± 0.9 | 38.7 ± 1.0 | | | | | 28.8 ± 1.3 | 27.9 ± 1.0 | 72.2 ± 0.7 | 62.2 ± 0.8 | | | | | 57.4 ± 0.9 | 61.7 ± 1.0 |
| **MSSS [30]** | 65.7 ± 0.8 | 33.6 ± 1.0 | | | | | 30.9 ± 1.1 | 31.0 ± 1.0 | 73.5 ± 0.6 | 56.5 ± 1.0 | | | | | 58.0 ± 0.9 | 57.8 ± 1.0 |
| **KFCM-CV [31]** | 83.6 ± 0.4 | | | | | 61.0 ± 1.0 | 56.1 ± 1.4 | 46.9 ± 1.2 | 87.1 ± 0.3 | | | | | 76.5 ± 0.7 | 68.5 ± 1.0 | 79.1 ± 0.8 |
| **FDE [34]** | 50.4 ± 0.9 | | | 22.8 ± 0.8 | | 22.0 ± 0.8 | 18.2 ± 1.1 | | 59.0 ± 1.0 | | | 50.8 ± 1.2 | | 46.4 ± 1.3 | 54.7 ± 1.1 | |
| **PAE [35]** | 62.3 ± 0.9 | | | 36.2 ± 1.0 | | 38.7 ± 1.0 | 38.5 ± 1.1 | | 73.2 ± 0.8 | | | 50.6 ± 1.5 | | 52.0 ± 1.4 | 53.4 ± 1.3 | |
| **MF [25]** | 79.1 ± 1.6 | | | | 37.4 ± 2.0 | 41.8 ± 1.2 | 48.2 ± 1.1 | | 88.0 ± 0.6 | | | | 61.8 ± 0.9 | 71.4 ± 1.1 | 65.1 ± 1.2 | |
| **TE [32]** | 78.2 ± 0.6 | | 31.8 ± 0.8 | | | | 39.5 ± 1.2 | 38.2 ± 1.4 | 80.5 ± 1.0 | | 63.3 ± 1.1 | | | | 52.8 ± 1.3 | 61.5 ± 1.6 |
| **FICA [33]** | 71.4 ± 1.3 | | 33.5 ± 1.6 | | | | 35.4 ± 1.7 | 41.6 ± 1.3 | 88.5 ± 0.5 | | 74.4 ± 1.2 | | | | 70.1 ± 1.1 | 73.0 ± 0.9 |

**Notations**: M0: No post-processing; M1: FiltDil; M2: HeurFilt; M3: 3Stage scheme; M4: S&O; M5: S&C; M6: Enh3Stage scheme; M7: SpatialProc. The same notations are adopted in Table 6.

**Table 6.** MODA and MOC for ten algorithms before and after combining with their best three post-processing schemes.

| Metric | Alg. \ Scheme | Tucson Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **LC [27]** | **VMO [28]** | **FT [29]** | **MSSS [30]** | **KFCM-CV [31]** | **FDE [34]** | **PAE [35]** | **MF [25]** | **TE [32]** | **FICA [33]** |
| **MODA** | M0: No Post-Proc. | 0.295 ± 0.034 | −0.979 ± 0.098 | −0.713 ± 0.066 | −0.944 ± 0.070 | −3.944 ± 0.124 | 0.097 ± 0.031 | −0.531 ± 0.053 | −3.248 ± 0.359 | −2.402 ± 0.113 | −1.538 ± 0.158 |
| | M1: FiltDil [22] | | | 0.401 ± 0.025 | 0.523 ± 0.021 | | | | | | |
| | M2: HeurFilt [23] | | | | | | | | | 0.261 ± 0.032 | 0.564 ± 0.036 |
| | M3: 3Stage [13] | 0.684 ± 0.016 | 0.505 ± 0.018 | | | | 0.769 ± 0.009 | 0.585 ± 0.014 | | | |
| | M4: S&O [25] | | | | | | | | 0.602 ± 0.020 | | |
| | M5: S&C [4] | 0.664 ± 0.018 | 0.522 ± 0.021 | | | −0.160 ± 0.035 | 0.790 ± 0.008 | 0.510 ± 0.017 | 0.453 ± 0.023 | | |
| | M6: Enh3Stage [20] | | 0.563 ± 0.042 | 0.641 ± 0.075 | 0.602 ± 0.058 | 0.169 ± 0.041 | 0.814 ± 0.010 | 0.522 ± 0.013 | 0.307 ± 0.026 | 0.498 ± 0.017 | 0.582 ± 0.029 |
| | M7: SpatialProc [14] | 0.669 ± 0.021 | | 0.652 ± 0.016 | 0.610 ± 0.025 | 0.368 ± 0.042 | | | | 0.511 ± 0.022 | 0.428 ± 0.037 |
| **MOC** | M0: No Post-Proc. | 0.296 | −0.981 | −0.714 | −0.943 | −3.944 | 0.098 | −0.532 | −3.235 | −2.400 | −1.534 |
| | M1: FiltDil [22] | | | 0.401 | 0.523 | | | | | | |
| | M2: HeurFilt [23] | | | | | | | | | 0.262 | 0.565 |
| | M3: 3Stage [13] | 0.684 | 0.505 | | | | 0.770 | 0.585 | | | |
| | M4: S&O [25] | | | | | | | | 0.601 | 0.648 | 0.378 |
| | M5: S&C [4] | 0.664 | 0.522 | | | −0.160 | 0.790 | 0.510 | 0.453 | | |
| | M6: Enh3Stage [20] | | 0.570 | 0.638 | 0.596 | 0.167 | 0.808 | 0.516 | 0.299 | 0.492 | 0.575 |
| | M7: SpatialProc [14] | 0.672 | | 0.653 | 0.604 | 0.377 | | | | 0.505 | 0.433 |

**Table 6.** *Cont.*

| Metric | Alg. / Scheme | Phoenix Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **LC** | **VMO** | **FT** | **MSSS** | **KFCM-CV** | **FDE** | **PAE** | **MF** | **TE** | **FICA** |
| **MODA** | M0: No Post-Proc. | −0.280 ± 0.033 | −1.353 ± 0.103 | −1.130 ± 0.055 | −1.403 ± 0.060 | −4.473 ± 0.090 | −0.065 ± 0.034 | −1.293 ± 0.082 | −4.427 ± 0.220 | −2.302 ± 0.240 | −4.771 ± 0.225 |
| | M1: FiltDil [22] | | | −0.062 ± 0.026 | 0.140 ± 0.022 | | | | | | |
| | M2: HeurFilt [23] | | | | | | | | | 0.092 ± 0.050 | −0.514 ± 0.061 |
| | M3: 3Stage [13] | 0.105 ± 0.028 | 0.421 ± 0.021 | | | | 0.350 ± 0.023 | 0.316 ± 0.033 | | | |
| | M4: S&O [25] | | | | | | | | | 0.039 ± 0.026 | |
| | M5: S&C [4] | 0.020 ± 0.032 | 0.292 ± 0.031 | | | −0.837 ± 0.040 | 0.467 ± 0.022 | 0.258 ± 0.034 | -0.324 ± 0.049 | | |
| | M6: Enh3Stage [20] | | 0.382 ± 0.024 | 0.168 ± 0.017 | 0.150 ± 0.021 | −0.829 ± 0.045 | 0.271 ± 0.036 | 0.235 ± 0.028 | 0.044 ± 0.025 | 0.298 ± 0.019 | −0.211 ± 0.045 |
| | M7: SpatialProc [14] | 0.326 ± 0.021 | | 0.080 ± 0.026 | 0.113 ± 0.025 | −1.155 ± 0.062 | | | | 0.064 ± 0.037 | −0.453 ± 0.039 |
| **MOC** | M0: No Post-Proc. | −0.280 | −1.351 | −1.123 | −1.402 | −4.477 | −0.063 | −1.287 | −4.418 | −2.292 | −4.764 |
| | M1: FiltDil [22] | | | −0.061 | 0.141 | | | | | | |
| | M2: HeurFilt [23] | | | | | | | | | 0.094 | −0.514 |
| | M3: 3Stage [13] | 0.103 | 0.420 | | | | 0.352 | 0.319 | | | |
| | M4: S&O [25] | | | | | | | | 0.044 | | |
| | M5: S&C [4] | 0.019 | 0.291 | | | −0.838 | 0.468 | 0.261 | −0.321 | | |
| | M6: Enh3Stage [20] | | 0.379 | 0.171 | 0.146 | −0.436 | 0.268 | 0.239 | 0.146 | 0.302 | −0.207 |
| | M7: SpatialProc [14] | 0.330 | | 0.082 | 0.112 | −1.153 | | | | 0.065 | −0.452 |

Table 6 presents quantitative results of each algorithm without applying any post-processing or each combined with any of the voted post-processing schemes measured by MODA and MOC from CLEAR metric [41,42], where 95% CIs are applied for each metric.

From Table 6, we conclude that in the Tucson dataset, FDE with Enh3Stage scheme (M6) has highest MODA, KFCM-CV with SpatialProc (M7) shows largest improvement, KFCM-CV with S&C (M5) has lowest MODA, and LC with S&C (M5) has smallest improvement. In the Phoenix dataset, FDE with S&C (M5) shows highest MODA, FICA with Enh3Stage scheme has largest improvement, KFCM-CV with S&C (M5) shows lowest MODA, while LC with 3Stage scheme (M3) has smallest improvement. Checking the corresponding MOC indices of each method before and after combining with any of the three best post-processing schemes, every numerical value closely coincides with the related sample mean for each MODA.

#### 4.4.4. Qualitative Performance Evaluation

Figure 4 depicts a visual comparison of a salient region from the cropped subimages (size $64 \times 64$) of the 51st frame in Tucson and Phoenix datasets. This region contains the largest count of detected vehicles from each algorithm. From Figure 4, comparing the results without and with post-processing, we observe the successful removal of many FPs.

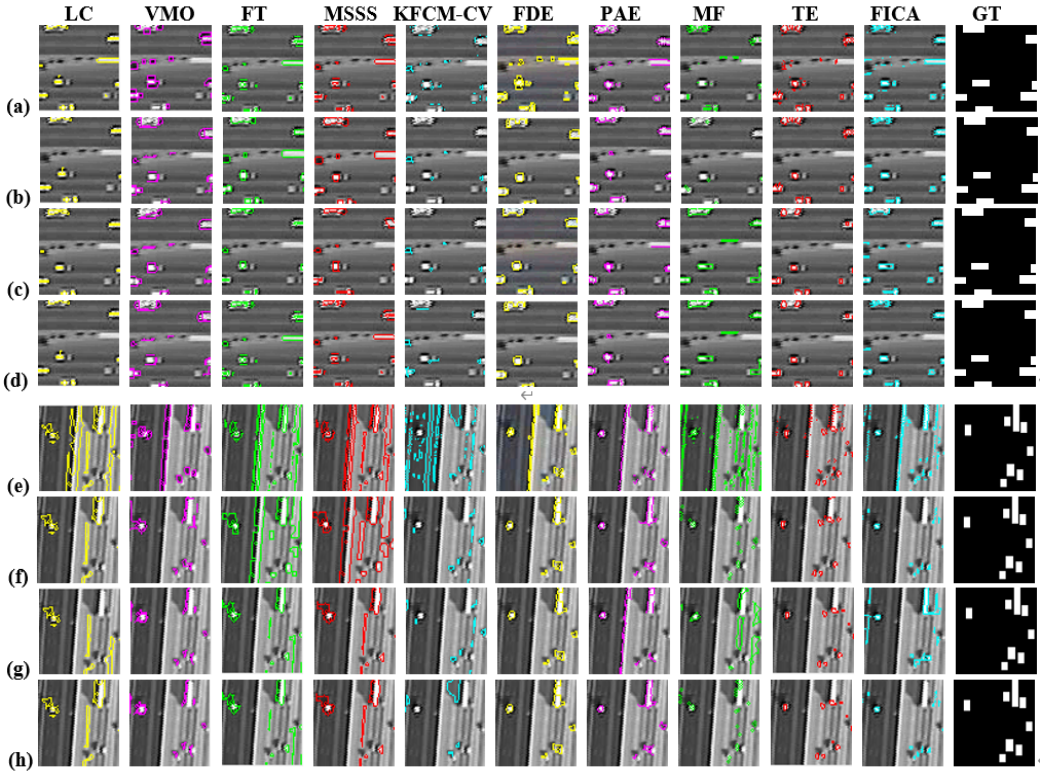


**Figure 4.** Visual comparison of ten algorithms without/with post-processing. Detections are shown as colored boundaries. Row (**a**) shows initial detection, (**b**–**d**) show results using the three post-processing schemes voted from Table 3 for Tucson dataset, respectively. Similar results are shown in row (**e**) and rows (**f**–**h**) for Phoenix dataset. GT vehicles are displayed in the last column.

#### 4.4.5. FC-DenseNet: Two-Stage Machine Learning for Small Object Detection

In this subsection, we present a third set of experiments to assess our two-stage machine learning approach (i.e., FC-DenseNet model and its variations) for small object detection. The limitations of our previous study and plan for solution are summarized as follows: (i) While our investigation included comprehensive tests on voting the best three post-processing schemes for small object detection using two low-resolution wide-area aerial datasets, the number of frames was relatively small, the results may not have universal applicability. Hence, the online VEDAI dataset with both training and testing folders were used for our additional experiments. (ii) In addition to saliency detection, the four "pillar" techniques on small object detection had been specified as multi-scale representation, contextual information, super resolution based techniques and regional proposals [1]; however, the ten algorithms [25,27–35] adapted for small object detection missed the involvement of regional proposals, i.e., deep CNN-based semantic image segmentation, where it was reported that convincible object detection accuracy could also be achieved on similar urban scene datasets such as CamVid and Gatech [39], and hence, this keynote approach should also be included for verification. (iii) Regarding the voted best three schemes (Enh3Stage, SpatialProc, S&C), subsequent competition is necessary upon handling a different online available aerial dataset; meanwhile, the other four schemes (filtered dilation, heuristic filtering, 3Stage, sieving and opening) with fewer votes in prior tests should not be completely excluded in another set of test scenarios when applying post-processing. As depicted in Figure 1, we performed tests to measure the degree of improvement from each post-processing scheme associated with the two-stage machine learning approach followed up with the updated FC-DenseNet. We tabulated the numerical results via a few sets of tests and expect to further evaluate the method by matching the best candidate to improve small object detection accuracy, applying a different ratio of training, validation and testing for online VEDAI dataset.

Some additional tests are presented as follows: (i) The vehicle tiramisu code [39] on semantic segmentation was implemented for the online VEDAI dataset (size: 512 × 512), where the portion of training data was progressively tuned from 80% to 95% in 5% steps then 98% as the highest (and the rest for testing), with a fixed portion of 1% (valid_pct = 0.01) for validation. (ii) The two-stage learning within ten epochs was applied to acquire the detection accuracy and draw the curve of learning rate versus loss, where some parameters such as weight decay (wd), learning rate (lr), and optimal threshold, were set up with default values and pct_start was initialized as 0.3/0.7. (iii) FC-DenseNet (with 103 layers and its other alternatives) was adopted upon using a relatively smaller learning rate (selected from the curve on the right plot of Figure 5) to proceed with the updated two-stage learning, where the final outputs including updated object detection accuracy with predicted detection labels. (iv) Final tests were designed to pick up four post-processing schemes (either from the voted three schemes and a relatively best one from the other four) to further evaluate the degree of improvement on the metrics of accuracy and mAP, then determine the best post-processing scheme (among the seven) for online VEDAI dataset [1,52,55].

```
# Start first-stage of learning
lr = 1e-3  # lr = 1e-3 as origin
# 0.3 is the default value for pct_start
learn.fit_one_cycle(10, slice(lr), pct_start = 0.3)
learn.save('stage-1')
learn.load('stage-1');
# learn.unfreeze()
learn.show_results(rows = 6, figsize=(8,20))
```

| epoch | train_loss | valid_loss | acc_camvid | time |
|---|---|---|---|---|
| 0 | 0.016666 | 0.016187 | 0.055480 | 44:16 |
| 1 | 0.015861 | 0.014790 | 0.055785 | 02:50 |
| 2 | 0.012056 | 0.012926 | 0.113250 | 02:46 |
| 3 | 0.009545 | 0.010552 | 0.238215 | 02:46 |
| 4 | 0.007233 | 0.008442 | 0.480341 | 02:46 |
| 5 | 0.005857 | 0.007016 | 0.488423 | 02:46 |
| 6 | 0.003914 | 0.005321 | 0.668588 | 02:46 |
| 7 | 0.002741 | 0.004416 | 0.692957 | 02:46 |
| 8 | 0.001947 | 0.004246 | 0.734986 | 02:46 |
| 9 | 0.001648 | 0.004192 | 0.736271 | 02:46 |

```
# Second stage of learning
learn.unfreeze()
lrs = slice(lr/400,lr/4)
learn.fit_one_cycle(10, lrs, pct_start = 0.3)

learn.save('stage-2');
learn.show_results(rows = 8, figsize = (6,20))
"""## Go big"""
```

| epoch | train_loss | valid_loss | acc_camvid | time |
|---|---|---|---|---|
| 0 | 0.001681 | 0.004435 | 0.717165 | 02:53 |
| 1 | 0.001685 | 0.004583 | 0.723914 | 02:52 |
| 2 | 0.001903 | 0.004001 | 0.729406 | 02:52 |
| 3 | 0.001466 | 0.003723 | 0.751148 | 02:51 |
| 4 | 0.001491 | 0.003677 | 0.783602 | 02:51 |
| 5 | 0.001257 | 0.002973 | 0.820088 | 02:50 |
| 6 | 0.001110 | 0.002958 | 0.796756 | 02:50 |
| 7 | 0.000927 | 0.003032 | 0.822462 | 02:51 |
| 8 | 0.000862 | 0.002777 | 0.833896 | 02:50 |
| 9 | 0.000829 | 0.002656 | 0.840210 | 02:51 |

```
wd = 1e-2
learn = unet_learner(data, models.resnet34, metrics=metrics

lr_find(learn)
learn.recorder.plot()
```

0.00% [0/1 00:0(

epoch train_loss valid_loss acc_camvid time

18.83% [58/308

LR Finder is complete, type {learner_name}.recorder.plot()

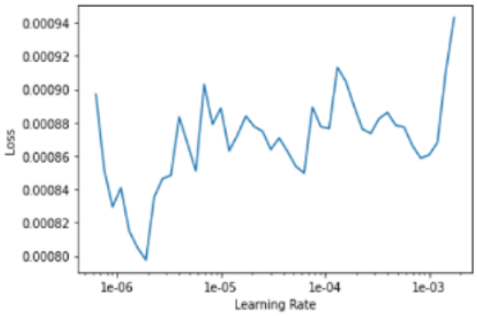


**Figure 5.** Sample results of two-stage learning approach and the generated curve to find an optimal learning rate (after parameter tuning) on semantic image segmentation for online VEDAI.

Regarding a CNN-related architecture of fully convolutional (FC) DenseNet with its dense block of four layers and its building blocks in [39], the differences in the connectivity pattern were established between each of the upsampling and downsampling paths, and the generation of feature maps was performed on each of the four layers and the block output from concatenation of each layer output, while the information of crucial parameters was excerpted from every layer of the model. Crucial kernels such as Transition Down and Transition Up were combined in the final output.

A subgroup of eight sample frames were loaded in the implementation of the two-stage learning approach for small object detection, where ten categories of objects may co-exist: terrain, car, truck, tractor, camping car, van, pickup, bus, boat, and plane. We presented the initial results of Google Colab program outputs after tuning each parameter in Figure 5, where we found that the highest accuracy achieved was 84.0%, and a better alternative learning rate (less than 0.001) close to the numerical point of 0.0001 appeared by the end of the second-stage learning.

The FC-DenseNet model [39] with 103 layers was applied to perform our updated two-stage learning, where the related parameters were selected as follows: wd = 0.01, lr = 0.0001, valid_pct = 0.01, and pct_start = 0.3/0.7. Figure 6 shows the updated results and

samples of predicted GT objects, where the highest accuracy was reached with 86.0% in the Epoch 6 of second-stage learning, and the predicted labels of small objects match the ground truth.

```
lr = 1e-4  # Updated First-stage Learning
# default value pct_start = 0.3
learn.fit_one_cycle(10, slice(lr), pct_start = 0.7)
learn.save('stage-1-big')
learn.load('stage-1-big');
learn.unfreeze()

learn.show_results(rows = 6, figsize=(8,20))
```

| epoch | train_loss | valid_loss | acc_camvid | time |
|---|---|---|---|---|
| 0 | 0.000789 | 0.002785 | 0.835626 | 02:45 |
| 1 | 0.000803 | 0.002864 | 0.830231 | 02:44 |
| 2 | 0.000804 | 0.003147 | 0.821219 | 02:44 |
| 3 | 0.000764 | 0.003023 | 0.827881 | 02:44 |
| 4 | 0.000786 | 0.003020 | 0.851042 | 02:44 |
| 5 | 0.000729 | 0.003125 | 0.846842 | 02:44 |
| 6 | 0.000692 | 0.003054 | 0.836376 | 02:45 |
| 7 | 0.000629 | 0.003004 | 0.848276 | 02:44 |
| 8 | 0.000558 | 0.003248 | 0.847903 | 02:44 |
| 9 | 0.000497 | 0.003248 | 0.848376 | 02:44 |

```
lr = 1e-4
lrs = slice(lr/1000,lr/10)  # Updated Second-stage Learning
learn.fit_one_cycle(10, lrs, pct_start = 0.7)
# learn.fit_one_cycle(10, lrs)
learn.save('stage-2-big')
learn.load('stage-2-big');

learn.show_results(rows = 10, figsize = (8,20))

"""## fin"""
```

| epoch | train_loss | valid_loss | acc_camvid | time |
|---|---|---|---|---|
| 0 | 0.000523 | 0.003173 | 0.855758 | 02:54 |
| 1 | 0.000484 | 0.003262 | 0.853271 | 02:54 |
| 2 | 0.000500 | 0.003299 | 0.851211 | 02:54 |
| 3 | 0.000508 | 0.003154 | 0.859316 | 02:53 |
| 4 | 0.000486 | 0.003281 | 0.852572 | 02:53 |
| 5 | 0.000482 | 0.003065 | 0.859772 | 02:53 |
| 6 | 0.000466 | 0.003187 | 0.860333 | 02:54 |
| 7 | 0.000481 | 0.003291 | 0.854467 | 02:54 |
| 8 | 0.000448 | 0.003338 | 0.850725 | 02:54 |
| 9 | 0.000448 | 0.003349 | 0.852821 | 02:55 |

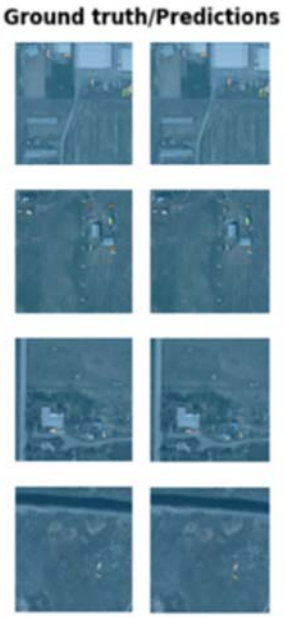


**Figure 6.** Sample results on detection accuracy and GT/predicted objects of two-stage learning after updating with FC-DenseNet 103 and applying an adjusted learning rate for online VEDAI.

Given the results obtained from the online VEDAI dataset as mentioned above, it is computed that the average detection accuracy is 78.2% on initial second stage learning and 85.5% on the updated second stage learning, proving FC-DenseNet 103 a better model on small object detection.

#### 4.4.6. Final Tests to Evaluate the Two-Stage Learning Approach with Post-Processing for Online VEDAI

Considering the low-resolution and wide area of online VEDAI dataset, we designed our final tests to select the best of the seven post-processing schemes. The experimental study was conducted in four scenarios. (i) Vote a scheme with highest score on accuracy among the other four post-processing schemes (with fewer votes from the first set of experiments). (ii) Take the vote from (i) as the fourth scheme along with the best three schemes, apply each post-processing scheme after updated two-stage learning, and evaluate the differences in the detection outputs, where accuracy and mAP are used as two metrics for quantitative comparison. (iii) Conduct sensitivity analysis to evaluate a balance between object detection accuracy, the complexity of algorithms, and time costs when applying each of the four post-processing schemes and varying the number of convolutional layers in FC-DenseNet. (iv) Combine the scores obtained from (i) to (iii) to determine the best post-processing scheme for online VEDAI. Note that the size and resolution of sample frames were adjusted upon applying any post-processing scheme.

Applying each of the four schemes (FiltDil (M1), HeurFilt (M2), 3Stage (M3), S&O (M4)) to post-process the detection output after updated two-stage learning, the scores of detection accuracy are shown in Table 7, where three different cases on ratio of training to testing on the online VEDAI dataset were considered. While applying M4 resulted in reducing accuracy, each of the other three schemes mildly increased their scores, and the relatively best results appeared with our 3Stage scheme (M3) in condition of choosing different portions of training to test.

**Table 7.** Performance evaluation of accuracy metric on each of the four unvoted post-processing schemes associated with initial detection after updated two-stage learning (FC-DenseNet 103).

| Training: Test | No Post-Proc. | FiltDil (M1) [22] | HeurFilt (M2) [23] | 3Stage (M3) [13] | S&O (M4) [25] |
|---|---|---|---|---|---|
| **80%:20%** | 0.718 | 0.767 | 0.746 | 0.771 | 0.682 |
| **90%:10%** | 0.745 | 0.790 | 0.772 | 0.798 | 0.706 |
| **95%:5%** | 0.801 | 0.853 | 0.829 | 0.856 | 0.754 |

We adopted the metric of mean average precision (mAP) on ten sample frames from online VEDAI to measure the post-processing results by FiltDil (M1) and 3Stage scheme (M3), where the results from five different portions of training to test are displayed in Table 8. We determined that the 3Stage scheme may perform better than filtered dilation since it has the least contrast (79.54% to 75.33%) of mAP in the last column. Hence, we may choose our 3Stage scheme (M3) as the fourth scheme in addition to the former voted three schemes (M5, M6 and M7) to compare the outcomes of post-processing for online VEDAI dataset.

**Table 8.** Mean average precision mAP (%) of ten sample frames when post-processed by FiltDil (M1) vs. 3Stage scheme (M3) after applying updated two-stage learning (FC-DenseNet 103).

| | 80%:20% | 85%:15% | 90%:10% | 95%:5% | 98%:2% |
|---|---|---|---|---|---|
| **No Post-Processing** | 63.90 ± 1.81 | 64.68 ± 1.65 | 67.14 ± 1.58 | 71.29 ± 1.43 | 74.18 ± 1.32 |
| **FiltDil (M1) [22]** | 65.96 ± 1.74 | 67.85 ± 1.92 | 68.73 ± 1.56 | 72.47 ± 1.69 | 75.33 ± 1.45 |
| **3Stage (M3) [13]** | 71.07 ± 1.25 | 72.98 ± 1.09 | 73.61 ± 1.28 | 77.89 ± 1.31 | 79.54 ± 1.16 |

Given the two metrics of accuracy and mAP to measure the improved performance of four post-processing methods, i.e., the 3Stage scheme (M3), S&C (M5), the Enh3Stage scheme (M6) and SpatialProc (M7) applied to the initial detection output after updated two-stage learning, we present the quantitative scores on each of the post-processing schemes in Table 9, where five different portions of training to test were applied in contrast with those same cases without any post-processing. When our training set is no greater than 90%, better scores were achieved by S&C (M5) than 3Stage (M3), while the opposite behavior occurs in conditions of a very large proportion of training (95% and 98%); the best improvement was achieved by Enh3Stage (M6), the least improvement was associated with SpatialProc (M7) among the four post-processing schemes using Accuracy and mAP as metrics. Hence, we conclude that among the seven post-processing schemes for the entire online VEDAI dataset, our Enh3Stage scheme (M6) achieved the best results: typically, when applying a training to test ratio of 98% to 2% for online VEDAI dataset, the highest mAP and accuracy were 82.80% and 89.1%, respectively.

**Table 9.** Mean average precision and accuracy when post-processed by each of the four schemes (M3, M5, M6 and M7) on the online VEDAI dataset.

| | | mAP (%) | | | |
|---|---|---|---|---|---|
| **Training: Test** | **No Post-Proc.** | **3Stage (M3) [13]** | **S&C (M5) [4]** | **Enh3Stage (M6) [20]** | **SpatialProc (M7) [14]** |
| **80%:20%** | 62.58 ± 1.73 | 69.84 ± 1.26 | 71.24 ± 1.68 | 75.32 ± 1.17 | 67.39 ± 1.52 |
| **85%:15%** | 64.29 ± 1.56 | 71.65 ± 1.12 | 73.08 ± 1.52 | 77.89 ± 1.06 | 69.23 ± 1.38 |
| **90%:10%** | 66.57 ± 1.48 | 72.98 ± 1.43 | 74.85 ± 1.36 | 79.24 ± 0.95 | 71.68 ± 1.25 |
| **95%:5%** | 70.92 ± 1.35 | 76.23 ± 0.97 | 75.86 ± 1.03 | 81.36 ± 0.88 | 74.36 ± 1.10 |
| **98%:2%** | 73.16 ± 1.29 | 78.45 ± 1.08 | 77.63 ± 0.94 | 82.80 ± 1.04 | 75.57 ± 0.99 |
| | | **Accuracy** | | | |
| **Training:Test** | **No Post-Proc.** | **3Stage (M3) [13]** | **S&C (M5) [4]** | **Enh3Stage (M6) [20]** | **SpatialProc (M7) [14]** |
| **80%:20%** | 0.718 | 0.771 | 0.780 | 0.823 | 0.742 |
| **85%:15%** | 0.729 | 0.783 | 0.791 | 0.844 | 0.756 |
| **90%:10%** | 0.745 | 0.798 | 0.817 | 0.865 | 0.773 |
| **95%:5%** | 0.801 | 0.856 | 0.835 | 0.882 | 0.808 |
| **98%:2%** | 0.846 | 0.874 | 0.858 | 0.891 | 0.835 |

#### 4.4.7. Computational Efficiency

Computational efficiency of each algorithm combined with its best post-processing scheme was evaluated for the involved dataset(s). Experiments were conducted in the software platform of MATLAB R2019b on a Dell laptop with Intel Core i7-8500U 1.80 GHz CPU and 16 GB RAM. The average CPU execution times in seconds per frame with size 720 × 480 for eleven algorithms are reported in Table 10, indicating that among the eleven detection algorithms using their related aerial video dataset(s) for the experimental study, after combining the best post-processing scheme, FDE runs the fastest, FC-DenseNet is at the median position, and TE is the slowest in this experiment.

**Table 10.** Average CPU execution time (s) per frame for each of the eleven algorithms combined with the finally voted Enh3Stage post-processing scheme.

| Alg. | LC [27] | VMO [28] | FT [29] | MSSS [30] | KFCM-CV [31] | FDE [34] | PAE [35] | MF [25] | TE [32] | FICA [33] | FC-DenseNet [39] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Run Time** | 2.06 | 8.36 | 2.18 | 5.24 | 10.97 | 1.28 | 3.15 | 1.63 | 30.22 | 10.46 | 4.43 |

## 5. Discussion

We have presented a method for selecting post-processing schemes for the eleven automatic vehicle detection algorithms in low-resolution wide-area aerial imagery. In addition to the four existing post-processing schemes [22–25] and the S&C scheme which comprises pixel-area sieving and morphological closing [4], three more post-processing schemes we recently derived were included [13,14,20]. The 3Stage scheme [13] displays a better average F-score than filtering by shape index [24] for LC [27], VMO [28], FT [29], MSSS [30], KFCM-CV [31], and TE [32]. Our tests applied to two aerial datasets, i.e., Tucson and Phoenix, which take the average F-score as a metric for comparison on all frames of all combinations of ten algorithms with each of the seven post-processing schemes. Voting from highest average F-scores by row comparison, the best three post-processing schemes were respectively associated with each detection algorithm. The highest number of votes was established for Enh3Stage scheme (M6) [20], while the second highest votes was tied for S&C (M5) [4] and SpatialProc (M7). We conclude that after post-processing, in the Tucson dataset, FDE and LC rank the best two in precision, F-score and PWC; FT and MSSS rank the best two in recall. For the Phoenix dataset, FDE and VMO rank the best two in precision, F-score and PWC; VMO and MSSS rank the best two in recall. The metrics of MODA and MOC show coincidence with ranks of each automatic algorithm on PWC score improvements. The count of votes suggests possible outcome of improving the accuracy of small object detection in low-resolution aerial image datasets.

The hundred-layer vehicle tiramisu code [39] on semantic image segmentation was adapted as an eleventh algorithm on small object detection to design the third set of experiments in our research study. The two-stage machine learning approach [39] and its updated two-stage learning applying FC-DenseNet103 model with tuned parameters, were used to achieve the highest overall initial detection accuracy, where the best average accuracy was 85.5%. Two sets of experiments were designed to search for the best post-processing scheme between the two groups: applying mean average precision (mAP) and accuracy as two metrics for performance evaluation, we checked the other four schemes to determine the best one, and let this scheme join the group with the best three schemes voted from our first set of experiments. At the final stage, we conclude that the Enh3Stage scheme (M6), has the highest overall mAP and accuracy among the four post-processing schemes (M3, M5, M6 and M7) when different proportions of training and test images were chosen from online VEDAI dataset. Since the best outcome from Enh3Stage was just slightly lower than 85.0% on mAP and very close to 0.9 on accuracy, there are still some opportunities for future improvement.

There are several limitations of our research study. (i) Many feature-extraction-based algorithms on aerial vehicle detection are geometric measure-based or grayscale intensity-based methods, which lack temporal analysis. (ii) For fair comparison, all seven post-

processing schemes provide heuristic improvement for each of the grayscale aerial frames, where false positives may be further eliminated using temporal filtering in aerial video datasets. (iii) Since there was no training data for the two aerial video datasets, our study did not assess machine learning schemes on those datasets. (iv) Apply some newer metrics, i.e., the structure similarity index measure (SSIM) [60] for performance evaluation. (v) To demonstrate the robustness of the algorithms, many large datasets are needed for testing. All these topics may represent potential research directions for subsequent investigation.

Most recently, quite a few deep-learning approaches have been widely applied to object detection and segmentation in traffic video analysis as well as wide-area remote surveillance [57,61–75]. Detecting aerial vehicles using a deep learning scheme typically includes pretraining, sample frame labeling, feature extraction in a deep convolutional neural network (DCNN), and then implementation of an algorithm for object detection and segmentation to obtain the labeled outputs. While some multi-task learning-based methods have demonstrated time efficiency and object detection accuracy [43–45], further updates are still needed to deal with the increased complexity, time cost, and requirement of multi-core GPU support.

## 6. Conclusions

Our work addresses the accurate detection of small objects for vehicle detection using low-resolution wide-area datasets. We designed three sets of experiments for selecting post-processing schemes to improve the performance of object detection algorithms, In the first set of experiments, we voted the best three post-processing schemes combined with ten detection algorithms. In the second set of experiments, we determined the best post-processing scheme for each of the ten algorithms, based on the type of detections and two sets of performance metrics. In the third set of experiments, we applied these post-processing schemes to a two-stage machine learning approach and its variation model adding kernel of FC-DenseNet, and then measured the degree of improvement for online VEDAI. We quantified the object detection performance using basic IR metrics and CLEAR metrics. Based on average F-score, accuracy and mAP, we determined that the Enh3Stage scheme may represent the best scheme in our post-processing selection for improving vehicle detection accuracy in wide-area aerial imagery.

As future work, we propose to implement and improve some recently developed automatic detection schemes [7–13,20,55–57,62–70,75] for vehicle detection in wide-area aerial imagery. With reference to some other visual models on feature-based front-vehicle detection [75] and some information security-related classification schemes such as real-time threat detection [76], quasi-cliques analysis [77], and anomaly behavior analysis [78,79]; we also intend to develop a 3D motion filter with reference to some latest deep-learning approaches [58,60–62,64–67,71,80–85] for the accurate detection of small objects including aerial vehicle detection [47,86–95], tiny target visualization and recognition [1,7–9,14, 20,33,38,52,55,56,61–63,67–69,75,96–99] for the remotely monitored aerial video surveillance. Besides, we propose to exploit common features, i.e., geometry, orientation, and grayscale intensity. In addition, we need to improve the spatio-temporal processing scheme [14,20,81] by considering other hybrid approaches in the computer vision field, i.e., multi-scale structure information [6], multiple instance learning [100], and sparse representations [71,101–103]. While employing any subset of methods from unsupervised, semi-supervised, and multi-task feature learning strategies [64,69,75,96,104–108] on object detection, classification, and recognition, broader practical applications in the remote sensing domain may benefit from our research study.

**Author Contributions:** Conceptualization, X.G.; methodology, X.G., S.R. and S.S.; software, X.G.; validation, X.G., S.R. and R.C.P.; formal analysis, X.G.; investigation, X.G.; resources, X.G., S.R. and J.J.R.; data curation, X.G. and S.R.; writing—original draft preparation, X.G.; writing—review and editing, X.G., S.R., J.J.R., J.S., P.S., J.P. and S.H.; visualization, X.G.; supervision, S.R., J.J.R., J.S., P.S., J.P. and S.H.; project administration, S.R. and P.S.; funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partly supported by the Air Force Office of Scientific Research (AFOSR) Dynamic Data-Driven Application Systems (DDDAS) award number FA9550-18-1-0427, National Science Foundation (NSF) research projects NSF-1624668 and NSF-1849113, (NSF) DUE-1303362 (Scholarship-for-Service), National Institute of Standards and Technology (NIST) 70NANB18H263, and Department of Energy/National Nuclear Security Administration under Award Number(s) DE-NA0003946.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors are very grateful to anonymous reviewers for their suggestions on improving the quality of this manuscript. The authors also wish to thank Mark Hickman, School of Civil Engineering, at the University of Queensland, Australia, for providing some of the prior wide-area aerial video datasets.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chen, G.; Wang, H.-T.; Chen, K.; Li, Z.-Y.; Song, Z.-D.; Liu, Y.-L.; Knoll, A. A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, 1–18. [CrossRef]
2. Sivaraman, S.; Trivedi, M.M. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1773–1795. [CrossRef]
3. Gao, X.; Ram, S.; Rodríguez, J.J. A Performance Comparison of Automatic Detection Schemes in Wide-Area Aerial Imagery. In Proceedings of the 2016 IEEE Southwest Symposium of Image Analysis and Interpretation (SSIAI), Santa Fe, NM, USA, 6–8 March 2016; pp. 125–128.
4. Gao, X. Automatic Detection, Segmentation, and Tracking of Vehicles in Wide-Area Aerial Imagery. Master's Thesis, Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, USA, October 2016.
5. Prokaj, J.; Medioni, G. Persistent Tracking for Wide-Area Aerial Surveillance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1186–1193.
6. Ram, S.; Rodríguez, J.J. Vehicle Detection in Aerial Images using Multi-scale Structure Enhancement and Symmetry. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3817–3821.
7. Zhou, H.; Wei, L.; Creighton, D.; Nahavandi, S. Orientation aware vehicle detection in aerial images. *Electron. Lett.* **2017**, *53*, 1406–1408. [CrossRef]
8. Xiang, X.-Z.; Zhai, M.-L.; Lv, N.; Saddik, A.E. Vehicle counting based on vehicle detection and tracking from aerial videos. *Sensors* **2018**, *18*, 2560. [CrossRef]
9. Philip, R.C.; Ram, S.; Gao, X.; Rodríguez, J.J. A Comparison of Tracking Algorithm Performance for Objects in Wide Area Imagery. In Proceedings of the 2014 IEEE Southwest Symposium of Image Analysis and Interpretation (SSIAI), San Diego, CA, USA, 6–8 April 2014; pp. 109–112.
10. Han, S.-K.; Yoo, J.-S.; Kwon, S.-C. Real-time vehicle detection method in bird-view unmanned aerial vehicle imagery. *Sensors* **2019**, *19*, 3958. [CrossRef]
11. Gao, X. Vehicle detection in wide-area aerial imagery: Cross-association of detection schemes with post-processings. *Int. J. Image Min.* **2018**, *3*, 106–116. [CrossRef]
12. Gao, X. A thresholding scheme of eliminating false detections on vehicles in wide-area aerial imagery. *Int. J. Signal Image Syst. Eng.* **2018**, *11*, 217–224. [CrossRef]
13. Gao, X.; Ram, S.; Rodríguez, J.J. A post-processing scheme for the performance improvement of vehicle detection in wide-area aerial imagery. *Signal Image Video Process.* **2020**, *14*, 625–633, 635. [CrossRef]
14. Gao, X.; Szep, J.; Satam, P.; Hariri, S.; Ram, S.; Rodríguez, J.J. Spatio-temporal processing for automatic vehicle detection in wide-area aerial video. *IEEE Access* **2020**, *8*, 199562–199572. [CrossRef]
15. Porter, R.; Fraser, A.M.; Hush, D. Wide-area motion imagery. *IEEE Signal Process. Mag.* **2010**, *27*, 56–65. [CrossRef]
16. Tong, K.; Wu, Y.-Q.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]
17. Schilling, H.; Bulatov, D.; Niessner, R.; Middelmann, W.; Soergel, U. Detection of vehicles in multisensor data via multibranch convolutional neural networks. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2018**, *11*, 4299–4316. [CrossRef]
18. Shen, Q.; Liu, N.-Z.; Sun, H. Vehicle detection in aerial images based on lightweight deep convolutional network. *IET Trans. Image Process.* **2021**, *15*, 479–491. [CrossRef]
19. Wu, X.; Li, W.; Dong, D.-F.; Tian, J.-J.; Tao, R.; Du, Q. Vehicle detection of multi-source remote sensing data using active fine-tuning network. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 39–53. [CrossRef]

20. Gao, X. Performance evaluation of automatic object detection with post-processing schemes under enhanced measures in wide-are aerial imagery. *Multimed. Tools Appl.* **2020**, *79*, 30357–30386. [CrossRef]
21. Zhu, L.; Chen, J.-X.; Hu, X.-W.; Fu, C.-W.; Xu, X.-M.; Qin, J.; Heng, P.-A. Aggregating attentional dilated features for salient object detection. *IEEE Trans. Cir. Syst. Video Technol.* **2020**, *30*, 3358–3371. [CrossRef]
22. Salem, A.; Ghamry, N.; Meffert, B. Daubechies Versus Biorthogonal Wavelets for Moving Object Detection in Traffic Monitoring Systems. *Informatik-Berichte*. 2009, pp. 1–15. Available online: https://edoc.hu-berlin.de/bitstream/handle/18452/3139/229.pdf (accessed on 18 October 2021).
23. Samarabandu, J.; Liu, X.-Q. An edge-based text region extraction algorithm for indoor mobile robot navigation. *Int. J. Signal Process.* **2007**, *3*, 273–280.
24. Sharma, B.; Katiyar, V.K.; Gupta, A.K.; Singh, A. The automated vehicle detection of highway traffic images by differential morphological profile. *J. Transp. Technol.* **2014**, *4*, 150–156. [CrossRef]
25. Zheng, Z.-Z.; Zhou, G.-Q.; Wang, Y.; Liu, Y.-L.; Li, X.-W.; Wang, X.-T.; Jiang, L. A novel vehicle detection method with high resolution highway aerial image. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2013**, *6*, 2338–2343. [CrossRef]
26. Türmer, S. Car Detection in Low-Frame Rate Aerial Imagery of Dense Urban Areas. Ph.D. Thesis, Technische Universität München, Munich, Germany, 2014.
27. Mancas, M.; Gosselin, B.; Macq, B.; Unay, D. Computational Attention for Defect Localization. In Proceedings of the ICVS Workshop on Computational Attention and Applications, Bielefeld, Germany, 21–24 March 2007; pp. 1–10.
28. Saha, B.N.; Ray, N. Image thresholding by variational minimax optimization. *Pattern Recognit.* **2009**, *42*, 843–856. [CrossRef]
29. Achanta, R.; Hemami, S.; Estrada, F.; Süsstrunk, S. Frequency-Tuned Salient Region Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 1597–1604.
30. Achanta, R.; Süsstrunk, S. Saliency Detection using Maximum Symmetric Surround. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Hong Kong, China, 26–29 September 2010; pp. 2653–2656.
31. Wu, Y.-Q.; Hou, W.; Wu, S.-H. Brain MRI segmentation using KFCM and Chan-Vese model. *Trans. Tianjin Univ.* **2011**, *17*, 215–219. [CrossRef]
32. Huang, Z.-H.; Leng, J.-S. Texture extraction in natural scenes using region-based method. *J. Digital Inf. Manag.* **2014**, *12*, 246–254.
33. Ali, F.B.; Powers, D.M.W. Fusion-based fastICA method: Facial expression recognition. *J. Image Graph.* **2014**, *2*, 1–7. [CrossRef]
34. Gleason, J.; Nefian, A.V.; Bouyssounousse, X.; Fong, T.; Bebis, G. Vehicle detection from aerial imagery. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 2065–2070.
35. Trujillo-Pino, A.; Krissian, K.; Alemán-Flores, M.; Santana-Cedrés, D. Accurate subpixel edge location based on partial area effect. *Image Vis. Comput.* **2013**, *31*, 72–90. [CrossRef]
36. Ray, K.S.; Chakraborty, S. Object detection by spatio-temporal analysis and tracking of the detected objects in a video with variable background. *J. Visual Commun. Image Represent.* **2019**, *58*, 662–674. [CrossRef]
37. Tao, H.-J.; Lu, X.-B. Smoke vehicle detection based on multi-feature fusion and hidden Markov model. *J. Real-Time Image Process.* **2020**, *17*, 745–758. [CrossRef]
38. Tao, H.-J.; Lu, X.-B. Smoke vehicle detection based on spatiotemporal bag-of-features and professional convolutional neural network. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 3301–3316. [CrossRef]
39. Jégou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 11–19.
40. Shaikh, S.H.; Saeed, K.; Chaki, N. *Moving Object Detection Using Background Subtraction*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 30–31.
41. Kasturi, R.; Goldgof, D.; Soundararajan, P.; Manohar, V.; Garofolo, J.; Bowers, R.; Zhang, J. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 319–336. [CrossRef] [PubMed]
42. Kasturi, R.; Goldgof, D.; Soundararajan, P.; Manohar, V.; Boonstra, M.; Korzhova, V. Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (VACE-II). *Comput. Sci. Eng.* **2006**. Available online: https://catalog.ldc.upenn.edu/docs/LDC2011V03/ClearEval_Protocol_v5.pdf (accessed on 18 October 2021).
43. Li, Y.-Y.; Huang, Q.; Pei, X.; Chen, Y.-Q.; Jiao, L.-C.; Shang, R.-H. Cross-layer attention network for small object detection in remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2021**, *14*, 2148–2161. [CrossRef]
44. Li, C.-J.; Qu, Z.; Wang, S.-Y.; Liu, L. A method of cross-layer fusion multi-object detection and recognition based on improved faster R-CNN model in complex traffic environment. *Pattern Recognit. Lett.* **2021**, *145*, 127–134. [CrossRef]
45. Bosquet, B.; Mucientes, M.; Brea, V.M. STDnet-ST: Spatio-temporal ConvNet for small object detection. *Pattern Recognit.* **2021**, *116*, 107929. [CrossRef]
46. Wang, B.; Gu, Y.-J. An improved FBPN-based detection network for vehicles in aerial images. *Sensors* **2020**, *20*, 4709. [CrossRef] [PubMed]
47. Gao, P.; Tian, T.; Li, L.-F.; Ma, J.-Y.; Tian, J.-W. DE-CycleGAN: An object enhancement network for weak vehicle detection in satellite images. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2021**, *14*, 3403–3414. [CrossRef]
48. Rabbi, J.; Ray, N.; Schubert, M.; Chowdhury, S.; Chao, D. Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. *Remote Sens.* **2020**, *12*, 1432. [CrossRef]

49. Matczak, G.; Matzurek, P. Comparative Monte Carlo analysis of background estimation algorithms for unmanned aerial vehicle detection. *Remote Sens.* **2021**, *13*, 870. [CrossRef]
50. Li, S.; Zhou, G.-Q.; Zheng, Z.-Z.; Liu, Y.-L.; Li, X.-W.; Zhang, Y.; Yue, T. The Relation between Accuracy and Size of Structure Element for Vehicle Detection with High Resolution Highway Aerial Images. In Proceedings of the 2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Melbourne, Australia, 21–26 July 2013; pp. 2645–2648.
51. Elmikaty, M.; Stathaki, T. Car detection in aerial images of dense urban areas. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 51–63. [CrossRef]
52. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Visual Commun. Image Represent.* **2016**, *34*, 187–203. [CrossRef]
53. Nascimento, J.C.; Marques, J.S. Performance evaluation of object detection algorithms for video surveillance. *IEEE Trans. Multimed.* **2006**, *8*, 761–774. [CrossRef]
54. Karasulu, B.; Korukoglu, S. *Performance Evaluation Software: Moving Object Detection and Tracking in Videos*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; pp. 40–48.
55. Jiao, L.-C.; Zhang, F.; Liu, F.; Yang, S.-Y.; Li, L.-L.; Feng, Z.-X.; Qu, R. A survey of deep learning-based object detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
56. Chen, C.; Zhong, J.-D.; Tan, Y. Multiple oriented and small object detection with convolutional neural networks for aerial image. *Remote Sens.* **2019**, *11*, 2176. [CrossRef]
57. Shen, J.-Q.; Liu, N.-Z.; Sun, H.; Zhou, H.-Y. Vehicle detection in aerial images based on lightweight deep convolutional network and generative adversarial network. *IEEE Access* **2019**, *7*, 148119–148130. [CrossRef]
58. Qiu, H.-Q.; Li, H.-L.; Wu, Q.-B.; Meng, F.-M.; Xu, L.-F.; Ngan, K.N.; Shi, H.-C. Hierarchical context features embedding for object detection. *IEEE Trans. Multimedia* **2020**, *22*, 3039–3050. [CrossRef]
59. Boukerche, A.; Hou, Z.-J. Object detection using deep learning methods in traffic scenarios. *ACM Comput. Surv.* **2021**, *54*, 1–35. [CrossRef]
60. Fan, D.-P.; Cheng, M.-M.; Liu, Y.; Li, T.; Borji, A. Structure-measure: A New Way to Evaluate Foreground Maps. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4548–4557.
61. Chen, K.-Q.; Fu, K.; Yan, M.-L.; Gao, X.; Sun, X.; Wei, X. Semantic segmentation of aerial images with shuffling convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 173–177. [CrossRef]
62. Tayara, H.; Soo, K.-G.; Chong, K.-T. Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. *IEEE Access* **2018**, *6*, 2220–2230. [CrossRef]
63. Yang, J.-X.; Xie, X.-M.; Yang, W.-Z. Effective contexts for UAV vehicle detection. *IEEE Access* **2019**, *7*, 85042–85054. [CrossRef]
64. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3211–3231. [CrossRef]
65. Fu, Z.-H.; Chen, Y.-W.; Yong, H.-W.; Jiang, R.-X.; Zhang, L.; Hua, X.-S. Foreground gating and background refining network for surveillance object detection. *IEEE Trans. Image Process.* **2019**, *28*, 6077–6090. [CrossRef]
66. Franchi, G.; Fehri, A.; Yao, A. Deep morphological networks. *Pattern Recognit.* **2020**, *102*, 107246. [CrossRef]
67. Zhang, X.-X.; Zhu, X. Moving vehicle detection in aerial infrared image sequences via fast image registration and improved YOLOv3 network. *Int. J. Remote Sens.* **2020**, *41*, 4312–4335. [CrossRef]
68. Zhou, Y.-F.; Maskell, S. Detecting and Tracking Small Moving Objects in Wide Area Motion Imagery using Convolutional Neural Networks. In Proceedings of the 2019 22nd International Conference on Information Fusion, Ottawa, ON, Canada, 2–5 July 2019; pp. 1–8.
69. Yang, M.-Y.; Liao, W.-T.; Li, X.-B.; Rosenhahn, B. Deep-learning for Vehicle Detection in Aerial Images. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 3080–3084.
70. Liu, C.-Y.; Ding, Y.-L.; Zhu, M.; Xiu, J.-H.; Li, M.-Y.; Li, Q.-H. Vehicle detection in aerial images using a fast-oriented region search and the vector of locally aggregated descriptors. *Sensors* **2019**, *19*, 3294. [CrossRef]
71. Ram, S. Sparse Representations and Nonlinear Image Processing for Inverse Imaging Solutions. Ph.D. Thesis, Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, USA, 20 June 2017.
72. Huang, X.-H.; He, P.; Rangarajan, A.; Ranka, S. Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video. *ACM Trans. Spat. Alg. Syst.* **2020**, *6*, 1–28. [CrossRef]
73. Sommer, L.; Schuchert, T.; Beyerer, J. Comprehensive analysis of deep learning-based vehicle detection using aerial images. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2733–2747. [CrossRef]
74. Karim, S.; Zhang, Y.; Yin, S.-L.; Laghari, A.A.; Brohi, A.A. Impact of compressed and down-scaled training images on vehicle detection in remote sensing imagery. *Multimed. Tools Appl.* **2019**, *78*, 32565–32583. [CrossRef]
75. Song, J.-G.; Park, H.-Y. Object recognition in very low-resolution images using deep collaborative learning. *IEEE Access* **2019**, *7*, 134071–134082.
76. Shao, S.-C.; Tunc, C.; Satam, P.; Hariri, S. Real-time IRC Threat Detection Framework. In Proceedings of the 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W), Tucson, AZ, USA, 18–22 September 2017; pp. 318–323.

77. Bernard, J.; Shao, S.-C.; Tunc, C.; Kheddouci, H.; Hariri, S. Quasi-cliques' Analysis for IRC Channel Thread Detection. In Proceedings of the 7th International Conference on Complex Networks and Their Applications, Cambridge, UK, 11–13 December 2018; pp. 578–589.
78. Wu, C.-K.; Shao, S.-C.; Tunc, C.; Hariri, S. Video anomaly detection using pre-trained deep convolutional neural nets and context mining. In Proceedings of the IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), Antalya, Turkey, 2–5 November 2020; pp. 1–8.
79. Wu, C.-K.; Shao, S.-C.; Tunc, C.; Satam, P.; Hariri, S. An explainable and efficient deep learning framework for video anomaly detection. *Cluster Comput.* **2021**, *25*, 1–23. [CrossRef]
80. Kurz, F.; Azimi, S.-M.; Sheu, C.-Y.; D'Angelo, P. Deep learning segmentation and 3D reconstruction of road markings using multi-view aerial imagery. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 47. [CrossRef]
81. Yang, B.; Zhang, S.; Tian, Y.; Li, B.-J. Front-vehicle detection in video images based on temporal and spatial characteristics. *Sensors* **2019**, *19*, 1728. [CrossRef] [PubMed]
82. Mostofa, M.; Ferdous, S.N.; Riggan, B.S.; Nasrabadi, N.M. Joint-Srvdnet: Joint super resolution and vehicle detection network. *IEEE Access* **2020**, *8*, 82306–82319. [CrossRef]
83. Liu, Y.-J.; Yang, F.-B.; Hu, P. Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks. *IEEE Access* **2020**, *8*, 145740–145750. [CrossRef]
84. Qiu, H.-Q.; Li, H.-L.; Wu, Q.-B.; Shi, H.-C. Offset Bin Classification Network for Accurate Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 13188–13197.
85. Shao, S.-C.; Tunc, C.; Al-Shawi, A.; Hariri, S. Automated Twitter author clustering with unsupervised learning for social media forensics. In Proceedings of the IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
86. Lei, J.-F.; Dong, Y.-X.; Sui, H.-G. Tiny moving vehicle detection in satellite video with constraints of multiple prior information. *Int. J. Remote Sens.* **2021**, *42*, 4110–4125. [CrossRef]
87. Zhang, J.-P.; Jia, X.-P.; Hu, J.-K. Local region proposing for frame-based vehicle detection in satellite videos. *Remote Sens.* **2019**, *11*, 2372. [CrossRef]
88. Cao, S.; Yu, Y.-T.; Guan, H.-Y.; Deng, D.-F.; Yan, W.-Q. Affine-function transformation-based object matching for vehicle detection from unmanned aerial vehicle imagery. *Remote Sens.* **2019**, *11*, 1708. [CrossRef]
89. Fan, D.-P.; Ji, G.-P.; Cheng, M.-M.; Shao, L. Concealed object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1–17. [CrossRef]
90. Zhu, B.; Ao, K. Single-stage rotation-decoupled detector for oriented object. *Remote Sens.* **2020**, *12*, 3262.
91. Chen, J.; Wu, X.-X.; Duan, L.-X.; Chen, L. Sequential instance refinement for cross-domain object detection in images. *IEEE Trans. Image Process.* **2021**, *30*, 3970–3984. [CrossRef]
92. Ming, Q.; Miao, L.-J.; Zhou, Z.-Q.; Song, J.-J.; Yang, X. Sparse label assignment for oriented object detection in aerial images. *Remote Sens.* **2021**, *13*, 2664. [CrossRef]
93. Wu, J.-Q.; Xu, S.-B. From point to region: Accurate and efficient hierarchical small object detection in low-resolution remote sensing images. *Remote Sens.* **2021**, *13*, 2620. [CrossRef]
94. Zhang, T.; Tang, H.; Ding, Y.; Li, P.-L.; Ji, C.; Xu, P.-L. FSRSS-Net: High-resolution mapping of buildings from middle-resolution satellite images using a super-resolution semantic segmentation network. *Remote Sens.* **2020**, *13*, 2290. [CrossRef]
95. Zhou, S.-P.; Wang, J.-J.; Wang, L.; Zhang, J.-M.-Y.; Wang, F.; Huang, D.; Zheng, N.-N. Hierarchical and interactive refinement network for edge-preserving salient object detection. *IEEE Trans. Image Process.* **2021**, *30*, 1–14. [CrossRef] [PubMed]
96. Shao, S.-C.; Tunc, C.; Al-Shawi, A.; Hariri, S. An ensemble of ensembles approach to author attribution for internet relay chat forensics. *ACM Trans. Manage. Inform. Syst.* **2020**, *11*, 1–25. [CrossRef]
97. Shao, S.-C.; Tunc, C.; Al-Shawi, A.; Hariri, S. One-class Classification with Deep Autoencoder for Author Verification in Internet Relay Chat. In Proceedings of the IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
98. Zhu, L.-L.; Geng, X.; Li, Z.; Liu, C. Improving YOLOv5 with attention mechanism for detecting boulders from planetary images. *Remote Sens.* **2021**, *13*, 3776. [CrossRef]
99. Vasu, B.-K. Visualizing Resiliency of Deep Convolutional Network Interpretations for Aerial Imagery. Master's Thesis, Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA, December 2018.
100. Huang, F.; Qi, J.-Q.; Lu, H.-C.; Zhang, L.-H.; Ruan, X. Salient object detection via multiple instance learning. *IEEE Trans. Image Process.* **2017**, *26*, 1911–1922. [CrossRef]
101. Lu, H.-C.; Li, X.-H.; Zhang, L.-H.; Ruan, X.; Yang, M.-H. Dense and sparse reconstruction error-based saliency descriptor. *IEEE Trans. Image Process.* **2016**, *25*, 1592–1603. [CrossRef]
102. Liu, L.-C.; Chen, C.L.P.; You, X.-G.; Tang, Y.-Y.; Zhang, Y.-S.; Li, S.-T. Mixed noise removal via robust constrained sparse representation. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 2177–2189. [CrossRef]
103. Wang, H.-R.; Celik, T. Sparse representation based hyper-spectral image classification. *Signal. Image. Video Process.* **2018**, *12*, 1009–1017. [CrossRef]
104. He, Z.; Huang, L.; Zeng, W.-J.; Zhang, X.-N.; Jiang, Y.-X.; Zou, Q. Elongated small object detection from remote sensing images using hierarchical scale-sensitive networks. *Remote Sens.* **2021**, *13*, 3182. [CrossRef]

105. Wang, Y.; Jia, Y.-N.; Gu, L.-Z. EFM-Net: Feature extraction and filtration with mask improvement network for object detection in remote sensing images. *Remote Sens.* **2021**, *13*, 4151. [CrossRef]
106. Hou, Y.; Chen, M.-D.; Volk, R.; Soibelman, L. An approach to semantically segmenting building components and outdoor scenes based on multichannel aerial imagery datasets. *Remote Sens.* **2021**, *13*, 4357. [CrossRef]
107. Guo, Y.-L.; Wang, H.-Y.; Hu, Q.-Y.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3D point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [CrossRef] [PubMed]
108. Singh, P.S.; Karthikeyan, S. Salient object detection in hyperspectral images using deep background reconstruction based anomaly detection. *Remote Sens. Lett.* **2022**, *13*, 184–195. [CrossRef]